



Universidad  
Carlos III de Madrid

# **Generación Automática de Editores y Repositorios de Evidencias a partir de Modelos de Estándares de Seguridad**

---

**Trabajo Fin de Grado**

**Curso 2015-2016**

AUTOR: ELENA CORREAS MONTIEL

TUTOR: JOSÉ LUÍS DE LA VARA GONZÁLEZ

DEPARTAMENTO DE INFORMÁTICA

MADRID, JUNIO DE 2016

*Esta página se ha dejado en blanco de manera  
intencionada.*

**Título:** Generación automática de editores y repositorios de evidencias a partir de modelos de estándares de seguridad

**Autor:** Elena Correás Montiel

**Tutor:** José Luis de la Vara González

## **TRIBUNAL**

**Presidente:** Javier Carbo Rubiera

**Secretario:** Eduardo Herránz Sánchez

**Vocal:** Raúl Jiménez Recaredo

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día 4 de Julio de 2016 en el campus de Colmenarejo de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACION de \_\_\_\_\_

## **VOCAL**

**SECRETARIO**

**PRESIDENTE**

*Esta página se ha dejado en blanco de manera  
intencionada.*

# Contenido del documento

## Índice de contenido.

<b>Contenido del documento .....</b>	<b>5</b>
Índice de tablas. ....	8
Índice de ilustraciones. ....	12
<b>Summary.....</b>	<b>15</b>
Introduction .....	15
Goals.....	16
Development and results of the plugin.....	20
Development and results of the application .....	21
Conclusions .....	24
Total cost of the development .....	25
<b>1.    Introducción .....</b>	<b>26</b>
1.1    Motivación .....	26
1.2    Objetivos.....	27
1.3    Método de Resolución.....	27
1.4    Organización del Documento .....	28
1.5    Definiciones, Acrónimos y Abreviaturas .....	29
1.5.1    Definiciones .....	29
1.5.2    Acrónimos y abreviaturas.....	30
1.6    Aclaraciones sobre el documento.....	30
<b>2.    Análisis de situación.....</b>	<b>32</b>
2.1    Entorno Socio-Económico .....	32
2.2    Marco regulador.....	33
2.3    Sistemas críticos de seguridad.....	35
2.4    Certificación de sistemas críticos de seguridad .....	35
2.5    Estándares para sistemas críticos de seguridad.....	36
2.6    Gestión de evidencias de seguridad.....	37
2.7    Ingeniería dirigida por modelos.....	38
2.8    Modelos para la certificación de sistemas críticos de seguridad .....	38

2.9	Herramientas para la certificación de sistemas críticos de seguridad .....	42
2.10	Diferencias entre las aplicaciones existentes y la implementada .....	46
<b>3.</b>	<b>Análisis del Proyecto .....</b>	<b>47</b>
3.1	Casos de uso.....	47
3.1.1	Definición.....	47
3.1.2	Modo de exposición .....	48
3.1.3	Casos de Uso.....	49
3.2	Requisitos .....	54
3.2.1	Propósito y alcance.....	54
3.2.2	Visión General .....	54
3.2.3	Modelo de exposición de los requisitos .....	54
3.2.4	Requisitos funcionales.....	56
3.2.5	Requisitos de Interfaz.....	73
3.2.5.1	Interfaces de los Usuarios .....	73
3.2.5.2	Interfaces Software .....	76
3.2.5.3	Interfaces Hardware.....	76
3.2.6	Requisitos no funcionales.....	77
3.2.6.1	Requisitos No Funcionales de Rendimiento.....	77
3.2.6.2	Requisitos No Funcionales de Fiabilidad .....	78
3.2.6.3	Requisitos No Funcionales de Disponibilidad .....	79
3.2.6.4	Requisitos No Funcionales de Portabilidad.....	79
3.3	Trazabilidad entre Casos de Uso y Requisitos Funcionales.....	80
<b>4.</b>	<b>Diseño e Implementación.....</b>	<b>82</b>
4.1	Alternativas y decisiones de implementación .....	82
4.1.1	Alternativas estudiadas .....	82
4.1.1.1	Alternativas para la creación del editor .....	82
4.1.1.2	Alternativas de almacenamiento .....	83
4.1.1.3	Alternativas de diseño.....	83
4.1.2	Arquitectura software .....	84
4.2	Implementación de la aplicación .....	84
4.2.1	Editor gráfico .....	85
4.2.2	Almacenamiento .....	87
4.2.3	Diseño.....	93
4.2.3.1	Pantalla Principal.....	93
4.2.3.2	Pantalla de Manipulación de Artefactos .....	94

4.2.3.3	Pantalla de Manipulación de Procesos .....	95
4.2.3.4	Pantalla de Manipulación de Requisitos .....	97
4.2.3.5	Pantalla de Monitorización del proyecto .....	98
4.2.4	Organización del proyecto.....	99
4.2.5	Diagrama de clases .....	100
4.2.6	Diagramas de secuencia .....	122
4.2.6.1	Caso de Uso: Crear Proyecto .....	122
4.2.6.2	Caso de Uso: Monitorizar Progreso .....	123
4.2.6.3	Caso de Uso: Gestionar Artefactos .....	125
4.2.6.4	Caso de Uso: Gestionar Información de procesos .....	126
4.2.6.5	Caso de Uso: Gestionar Información de Justificación .....	128
<b>5.</b>	<b>Uso de la Aplicación .....</b>	<b>129</b>
5.1	Creación del Reference Assurance Framework .....	129
5.2	Requisitos de instalación previos al uso de la aplicación .....	134
5.2.1	Archivos de configuración .....	134
5.3	Guía de uso de la aplicación .....	135
<b>6.</b>	<b>Pruebas de Sistema .....</b>	<b>144</b>
6.1	Formato de las pruebas .....	144
6.2	Pruebas llevadas a cabo .....	145
6.3	Estado de las pruebas de sistema y trazabilidad .....	183
<b>7.</b>	<b>Gestión del Proyecto .....</b>	<b>187</b>
7.1	Planificación .....	187
7.1.1	Diagrama de Gantt .....	188
7.2	Presupuesto .....	190
7.2.1	Coste de Personal .....	190
7.2.2	Coste de Material .....	191
7.2.3	Coste total asociado al proyecto .....	192
<b>8.</b>	<b>Conclusiones y Líneas Futuras .....</b>	<b>193</b>
8.1	Conclusiones .....	193
8.2	Líneas Futuras .....	194
<b>9.</b>	<b>Bibliografía .....</b>	<b>196</b>

## Índice de tablas.

Tabla 1: Tool support summary. ....	18
Tabla 2: Total cost of the project. ....	25
Tabla 3: Funcionalidades implementadas por las herramientas similares. ....	46
Tabla 4: Formato de los casos de uso. ....	48
Tabla 5: Caso de uso "Crear Reference Assurance Framework".....	49
Tabla 6: Caso de uso "Crear Proyecto" .....	50
Tabla 7: Caso de uso "Monitorizar Progreso" .....	51
Tabla 8: Caso de Uso "Gestionar Artefactos" .....	52
Tabla 9: Caso de Uso "Gestionar Información de Proceso.....	53
Tabla 10: Caso de Uso "Gestionar Información de Justificación" .....	54
Tabla 11: Formato de los requisitos .....	55
Tabla 12: Requisito Funcional "Pantalla inicial de la aplicación" .....	56
Tabla 13: Requisito Funcional "Selección de proyectos" .....	56
Tabla 14: Requisito Funcional "Añadir un nuevo proyecto" .....	57
Tabla 15: Requisito Funcional "Creación de nuevos Artifact Definitions" .....	57
Tabla 16: Requisito Funcional "Añadir nuevos artefactos" .....	57
Tabla 17: Requisito Funcional "Modificación de Artifact Definitions" .....	58
Tabla 18: Requisito Funcional "Modificación de los artefactos" .....	58
Tabla 19: Requisito Funcional "Borrar Artifact Definitions " .....	59
Tabla 20: Requisito Funcional "Borrar artefactos" .....	59
Tabla 21: Requisito Funcional "ver detalles de un Artifact Definition" .....	59
Tabla 22: Requisito Funcional "ver detalles de un artefacto" .....	60
Tabla 23: Requisito Funcional "Añadir eventos." .....	60
Tabla 24: Requisito Funcional "Tipos de eventos" .....	60
Tabla 25: Requisito Funcional "Añadir evaluación." .....	60
Tabla 26: Requisito Funcional "Creación de técnicas" .....	61
Tabla 27: Requisito Funcional "Modificación de técnicas" .....	61
Tabla 28: Requisito Funcional "Borrado de técnicas" .....	62
Tabla 29: Requisito Funcional "ver detalles de una técnica" .....	62
Tabla 30: Requisito Funcional "añadir asociaciones entre artefactos y técnicas" .....	62
Tabla 31: Requisito Funcional "modificar asociaciones entre artefactos y técnicas" .....	63
Tabla 32: Requisito Funcional "Creación de roles" .....	63
Tabla 33: Requisito Funcional "Modificación de roles" .....	64



Tabla 34: Requisito Funcional "Borrado de roles" .....	64
Tabla 35: Requisito Funcional "ver detalles de una técnica" .....	64
Tabla 36: Requisito Funcional "añadir asociaciones entre artefactos y roles" .....	65
Tabla 37: Requisito Funcional "modificar asociaciones entre artefactos y roles" .....	65
Tabla 38: Requisito Funcional "Creación de actividades" .....	65
Tabla 39: Requisito Funcional "Modificación de actividades" .....	66
Tabla 40: Requisito Funcional "Borrado de actividades" .....	66
Tabla 41: Requisito Funcional "ver detalles de una actividad" .....	67
Tabla 42: Requisito Funcional "añadir asociaciones entre artefactos y actividades" .....	67
Tabla 43: Requisito Funcional "modificar asociaciones entre artefactos y actividades" .....	67
Tabla 44: Requisito Funcional "Creación de requisitos" .....	68
Tabla 45: Requisito Funcional "Modificación de requisitos" .....	68
Tabla 46: Requisito Funcional "Borrado de requisitos" .....	68
Tabla 47: Requisito Funcional "ver detalles de un requisito" .....	69
Tabla 48: Requisito Funcional "añadir asociaciones entre artefactos y requisitos" .....	69
Tabla 49: Requisito Funcional "modificar asociaciones entre artefactos y requisitos" .....	70
Tabla 50: Requisito Funcional "Ver progreso del proyecto" .....	70
Tabla 51: Requisito Funcional "Creación de un nuevo Reference Assurance Framework" .....	70
Tabla 52: Requisito Funcional "Modificación del Reference Assurance Framework" .....	71
Tabla 53: Requisito Funcional "Añadir nuevos Reference Artifacts al modelo" .....	71
Tabla 54: Requisito Funcional "Añadir nuevos Reference Techniques al modelo" .....	71
Tabla 55: Requisito Funcional "Añadir nuevos Reference Roles al modelo" .....	71
Tabla 56: Requisito Funcional "Añadir nuevos Reference Activities al modelo" .....	72
Tabla 57: Requisito Funcional "Añadir nuevos Reference Requirements al modelo" .....	72
Tabla 58: Requisito Funcional "Añadir nuevos Reference Attributes al modelo" .....	72
Tabla 59: Requisito Funcional "Añadir nuevas relaciones entre los Reference Artifacts" .....	73
Tabla 60: Requisito Funcional "Borrado del Reference Assurance Framework" .....	73
Tabla 61: Requisito de Interfaz de Usuario "Un requisito por línea" .....	74
Tabla 62: Requisito de Interfaz de Usuario "Fondo de la interfaz" .....	74
Tabla 63: Requisito de Interfaz de Usuario "Botones de navegación" .....	74
Tabla 64: Requisito de Interfaz de Usuario "Idioma" .....	74
Tabla 65: Requisito de Interfaz de Usuario "Botones de añadir y edición" .....	75
Tabla 66: Requisito de Interfaz de Usuario "Estandarización de la interfaz" .....	75
Tabla 67: Requisito de Interfaz de Usuario "Logo" .....	75
Tabla 68: Requisito de Interfaz de Usuario "Botón guardar" .....	75

Tabla 69: Requisito de Interfaces Software "Conexión a base de datos" .....	76
Tabla 70: Requisito de Interfaces Software "Servidor" .....	76
Tabla 71: Requisito de Interfaces hardware "Resolución de pantalla" .....	76
Tabla 72: Requisito de Interfaces Hardware "Compatibilidad con navegadores" .....	77
Tabla 73: Requisito No Funcional de Rendimiento "Usuarios simultáneos" .....	77
Tabla 74: Requisito No Funcional de Rendimiento "Tiempo de reacción" .....	77
Tabla 75: Requisito No Funcional de Rendimiento "Comunicación con la base de datos" .....	78
Tabla 76: Requisito No Funcional de Rendimiento "Tiempo de navegación" .....	78
Tabla 77: Requisito No Funcional de Fiabilidad "Técnicas de prevención de errores SW" .....	78
Tabla 78: Requisito No Funcional de Disponibilidad "Disponibilidad de la aplicación" .....	79
Tabla 79: Requisito No Funcional de Portabilidad "Lenguaje utilizado en la aplicación" .....	79
Tabla 80: Requisito No Funcional de Portabilidad "Código independiente del servidor" .....	80
Tabla 81: Requisito No Funcional de Portabilidad "Lenguaje utilizado en la base de datos" .....	80
Tabla 82: Trazabilidad entre casos de uso y requisitos.....	81
Tabla 83: Prueba de sistema 1 .....	145
Tabla 84: Prueba de sistema 2 .....	145
Tabla 85: Prueba de sistema 3 .....	146
Tabla 86: Prueba de sistema 4 .....	146
Tabla 87: Prueba de sistema 5 .....	147
Tabla 88: Prueba de sistema 6 .....	147
Tabla 89: Prueba de sistema 7 .....	148
Tabla 90: Prueba de sistema 8 .....	148
Tabla 91: Prueba de sistema 9 .....	149
Tabla 92: Prueba de sistema 10 .....	150
Tabla 93: Prueba de sistema 11 .....	150
Tabla 94: Prueba de sistema 12 .....	151
Tabla 95: Prueba de sistema 13 .....	152
Tabla 96: Prueba de sistema 14 .....	152
Tabla 97: Prueba de sistema 15 .....	153
Tabla 98: Prueba de sistema 16 .....	154
Tabla 99: Prueba de sistema 17 .....	154
Tabla 100: Prueba de sistema 18 .....	155
Tabla 101: Prueba de sistema 19 .....	156
Tabla 102: Prueba de sistema 20 .....	156
Tabla 103: Prueba de sistema 21 .....	157

Tabla 104: Prueba de sistema 22 .....	158
Tabla 105: Prueba de sistema 23 .....	158
Tabla 106: Prueba de sistema 24 .....	159
Tabla 107: Prueba de sistema 25 .....	160
Tabla 108: Prueba de sistema 26 .....	160
Tabla 109: Prueba de sistema 27 .....	161
Tabla 110: Prueba de sistema 28 .....	162
Tabla 111: Prueba de sistema 29 .....	162
Tabla 112: Prueba de sistema 30 .....	163
Tabla 113: Prueba de sistema 31 .....	164
Tabla 114 : Prueba de sistema 32 .....	164
Tabla 115: Prueba de sistema 33 .....	165
Tabla 116: Prueba de sistema 34 .....	165
Tabla 117: Prueba de sistema 35 .....	166
Tabla 118: Prueba de sistema 36 .....	167
Tabla 119: Prueba de sistema 37 .....	167
Tabla 120: Prueba de sistema 38 .....	168
Tabla 121: Prueba de sistema 39 .....	169
Tabla 122: Prueba de sistema 40 .....	169
Tabla 123: Prueba de sistema 41 .....	170
Tabla 124: Prueba de sistema 42 .....	171
Tabla 125: Prueba de sistema 43 .....	171
Tabla 126: Prueba de sistema 44 .....	172
Tabla 127: Prueba de sistema 45 .....	172
Tabla 128: Prueba de sistema 46 .....	173
Tabla 129: Prueba de sistema 47 .....	174
Tabla 130: Prueba de sistema 48 .....	174
Tabla 131: Prueba de sistema 49 .....	175
Tabla 132: Prueba de sistema 50 .....	175
Tabla 133: Prueba de sistema 51 .....	175
Tabla 134: Prueba de sistema 52 .....	176
Tabla 135: Prueba de sistema 53 .....	176
Tabla 136: Prueba de sistema 54 .....	177
Tabla 137: Prueba de sistema 55 .....	177
Tabla 138: Prueba de sistema 56 .....	178

Tabla 139: Prueba de sistema 57 .....	178
Tabla 140: prueba de sistema 58 .....	178
Tabla 141: Prueba de sistema 59 .....	179
Tabla 142: Prueba de sistema 60 .....	179
Tabla 143: Prueba de sistema 61. ....	179
Tabla 144: Prueba de sistema 62 .....	180
Tabla 145: Prueba de sistema 63 .....	180
Tabla 146: Prueba de sistema 64 .....	180
Tabla 147: Prueba de sistema 65 .....	181
Tabla 148: Prueba de sistema 66 .....	181
Tabla 149: Prueba de sistema 67 .....	182
Tabla 150: Prueba de sistema 68 .....	182
Tabla 151: Prueba de sistema 69 .....	182
Tabla 152: Trazabilidad entre pruebas de sistema y requisitos funcionales. ....	186
Tabla 153: Resumen de las fases del proyecto. ....	188
Tabla 154: Coste de personal del proyecto.....	190
Tabla 155: Coste material del proyecto. ....	191
Tabla 156: Coste final del proyecto.....	192

## Índice de ilustraciones.

Ilustración 1: Metamodel for the graphical editor.....	20
Ilustración 2: Graphical editor appearance.....	21
Ilustración 3: Database schema. ....	22
Ilustración 4: Application organization. ....	23
Ilustración 5: Modelo del proceso de la certificación de las evidencias. ....	37
Ilustración 6: Modelo de SACM.....	39
Ilustración 7: Modelo de SafeTIM. ....	40
Ilustración 8: Modelo de IEC61508. ....	40
Ilustración 9: Modelo de RAF. ....	41
Ilustración 10: Meta-modelo utilizado para el desarrollo del proyecto. ....	42
Ilustración 11: Interfaz principal de LRDA. (LDRA).....	42
Ilustración 12: Interfaz Principal de GoedelWorks. (Altreonic) .....	43
Ilustración 13: Interfaz Principal de OPENCROSS (OPENCROSS) .....	43
Ilustración 14: Ejemplo de salida de interaz de CRESCO. (CRESCO, 2011).....	44

Ilustración 15: Esquema de uso de WEFACT. (WEFACT).....	44
Ilustración 16: Interfaz principal de PTC Integrity Process Director (Director).....	45
Ilustración 17: PTC Integrity Process Check (Check) .....	45
Ilustración 18: Diagrama de Casos de Uso .....	47
Ilustración 19: Esquema de la arquitectura Modelo-Vista-Controlador (V.Delgado, s.f.) .....	84
Ilustración 20: Meta-modelo utilizado para la generación del editor. ....	85
Ilustración 21: Pantalla de creación del GMF.....	85
Ilustración 22: Pantalla principal de creación del modelo. ....	86
Ilustración 23: Esquema relacional de la base de datos. ....	88
Ilustración 24: interfaz inicial de la pantalla principal.....	93
Ilustración 25: Interfaz final de la pantalla principal.....	94
Ilustración 26: Interfaz inicial de la pantalla de manipulación de artefactos. ....	94
Ilustración 27: Interfaz final de la pantalla principal de manipulación de artefactos.....	95
Ilustración 28: Diseño inicial de la pantalla de procesos .....	96
Ilustración 29: Interfaz final de la pantalla principal de manipulación de procesos.....	96
Ilustración 30: Diseño inicial de la pantalla de manipulación de requisitos. ....	97
Ilustración 31: Interfaz final de la pantalla de manipulación de requisitos. ....	98
Ilustración 32: Interfaz inicial de la pantalla de monitorización del proyecto.....	98
Ilustración 33: Interfaz final de la pantalla de monitorización del proyecto. ....	99
Ilustración 34: Organización del proyecto. ....	99
Ilustración 35: Diagrama de clases, parte I. ....	101
Ilustración 36: Diagrama de clases. Parte II. ....	104
Ilustración 37: Diagrama de clases. Parte III. ....	106
Ilustración 38: Diagrama de clases. Parte IV. ....	109
Ilustración 39: Diagrama de secuencia del caso de uso "Crear Proyecto".....	123
Ilustración 40: Diagrama de secuencia del caso de uso "Monitorizar Progreso" Primera parte. .....	124
Ilustración 41: Diagrama de secuencia del caso de uso "Monitorizar Progreso" Segunda Parte. .....	124
Ilustración 42: Diagrama de secuencia del caso de uso "Gestionar Artefactos". ....	126
Ilustración 43: Diagrama de secuencia "Gestionar informacion de proceso". ....	127
Ilustración 44: Diagrama de secuencia del caso de uso "Gestionar información de justificación". .....	128
Ilustración 45: Pantalla inicial del editor.....	129
Ilustración 46: Añadir un nuevo Reference Artefact.....	130

Ilustración 47: Añadir un nuevo atributo. ....	130
Ilustración 48: Añadir una nueva relación entre artefactos. ....	131
Ilustración 49: Creación de una nueva técnica. ....	131
Ilustración 50: Creación de un nuevo rol. ....	132
Ilustración 51: Creación de una nueva actividad. ....	133
Ilustración 52: Creación de un nuevo requisito. ....	133
Ilustración 53: Archivo configuration.properties ....	135
Ilustración 54: Archivo log4j.properties. ....	135
Ilustración 55: Pantalla inicial. ....	136
Ilustración 56: Pantalla de configuración. ....	137
Ilustración 57: Pantalla para introducir un nuevo Artifact Definition. ....	137
Ilustración 58: Pantalla para introducir un nuevo artefacto. ....	138
Ilustración 59: Pantalla para introducir una nueva técnica. ....	139
Ilustración 60: Pantalla para introducir un nuevo rol. ....	139
Ilustración 61: Pantalla para introducir una nueva actividad. ....	140
Ilustración 62: Pantalla de modificación de las actividades. ....	141
Ilustración 63: Pantalla para introducir un nuevo requisito. ....	141
Ilustración 64: Pantalla de modificación de los artefactos. ....	142
Ilustración 65: Pantalla para añadir un evento. ....	143
Ilustración 66: Pantalla para ver el progreso del proyecto. ....	143
Ilustración 67: Planificación de las tareas del proyecto. ....	189
Ilustración 68: Diagrama de Gantt ....	189

# Summary

The current document is a reporting of the bachelor's degree thesis "**Automatic generation of evidence editors and repositories from models of safety standards**". It contains the definition of the problem that will be solved, the goals to be achieved and the conclusions and solutions provided.

## Introduction

In recent years, intelligent applications and network connections have become more important each day. Therefore, and increasingly, there is a need to ensure application safety and data security, integrity and availability. The need to follow requirements and standards for safety, reliability and security makes the development of critical systems slower and more expensive than the rest of them.

Demonstration of compliance with established standards for these critical systems should be carried out a series of tests that, as they evolve, are becoming more robust.

Nowadays, Europe represents 30% of world production of embedded systems, with an advantage over other regions in industrial, automotive, medical and aerospace industries. The market for these systems is estimated at 850 M €. Within these figures, aerospace, automotive and medical sector as a whole represent a total of 472 M € worldwide, with the European figure of 141 M €. In the workplace, this type of system development each year generate approximately 50,000 jobs.

There are specific rules of the industries to be taken into account to develop tools for use in sectors considered critical. These are:

- **Railway industry:** ERA (*European Railway Agency*) is the responsible of the publication of standards. This sector is governed by the family of CENELEC standards.
- **Aerospace industry:** ESA (*European Space Agency*) is responsible of the publication of regulations to follow. This sector by the family of standards governing ECSS.

- **Aviation industry:** EASA (*European Aviation Safety Agency*) publishes the regulations related to this sector. It is governed by the family of DO standards (such as DO-178 standard) and ARP.
- **Medical devices:** In this case, each country has rules or independent certifications, but all have aspects in common. Despite that, there are also regulations at European level to be met, established by the EC.
- **Automotive:** Nowadays there is not a legal requirement, but many companies are starting to develop and assess products according to the ISO-26262 standard.
- **Other domains,** such as industrial or nuclear energy systems and oil & gas are also regulated by specific regulations.

## Goals

A critical system is one whose failure can cause very serious damage either to people or the environment. These critical systems in industries such as aviation, railway and automotive have to comply with stringent safety controls. The objective of the safety certification is to provide a formal endorsement of the particular system you want to operate safely against potential dangers that may arise. This certification can be associated with either assessment of products, processes or personnel.

Increasingly, different industries are governed by compliance with standards and use tools to support compliance-targeted activities. Such tools facilitate e.g. the storage of evidence to show that these standards are met, so that users can assure in case of possible problems in such critical systems.

The main objective of this work is to provide a tool that acts as editor and repository of evidence for safety standards, containing aspects that do not have other tools that are currently being used in various industries. Tool generation will be based on models.

The objectives to be met are the following:

- Create a graphic model editor for safety standards.
- Implement a tool that allows you to add evidence as set out in the models created.



- Monitor the progress of a certification project that the user is performing.
- Manage the evidence life cycle.

Some examples of similar evidence-based tools are:

- **LDRA**, verification tool of ISO 26262 standard on safety systems in cars. It is based on evidence and overall works as follows: the tool shows an ordered list of aspects to verify according to the standard, and the means to show compliance with these aspects.
- **GoedelWorks**, Altreonic's verifier against standards such as IEC61508, IEC62061, ISO26262, ISO13849, ISO25119, ISO15998, CMMI and "Automotive SPICE". It is implemented and available under the SaaS business model.
- **OPENCROSS**, based on evidences, has been developed in the scope of a large-scale European system dedicated to safety certification for markets in rail, aviation and automobile. It enables the choice of a reference certification model, supports the adaptation to specific standards, and records the life cycle of the evidence introduced.
- **CRESCO**, a tool for building repositories of evidence. To this end, a user draws a model to then generate a database schema according to the evidence added. The tool displays the evidence information through a web application, which also supports repository management.
- **WEFACT**, which facilitates the validation, verification and certification of safety systems in a modular way. The tool summarizes processes information and provides a basis for certification. It is possible to integrate external tools too.
- **PTC Integrity Process Director**, provides a practical approach to establish measures, management and improvement of the various operating processes of a company, focusing on compliance with standards. It is based on improved processes and real projects.

- **PTC Integrity Process Check**, responds to changes in the files, websites, and configuration management artifacts, among others. It includes checklists for the DO-178 and DO-254 standards.

The application has been developed in order to complete certain functions that existing tools do not possess today. The following table lists each functionality and indicates the support by the other tools. The way to be represented is:

- $\checkmark$ : meets functionality.
- - : Does not meet functionality completely, but slightly.
- X: Does not meet functionality.

	LDRA	GoedelWorks	OPENCOS	CRESCO	WEFACT	PTC Director	PTC Check
Life cycle management	-	-	$\checkmark$	-	-	-	-
Adaptation to standards	X	X	$\checkmark$	-	X	-	-
Derivation of the structure	X	X	X	$\checkmark$	X	X	X
Monitoring the state	$\checkmark$	X	$\checkmark$	X	X	$\checkmark$	$\checkmark$
Information Assurance	-	-	X	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

Tabla 1: Tool support summary.

Briefly, therefore, the application must implement the following functionality:

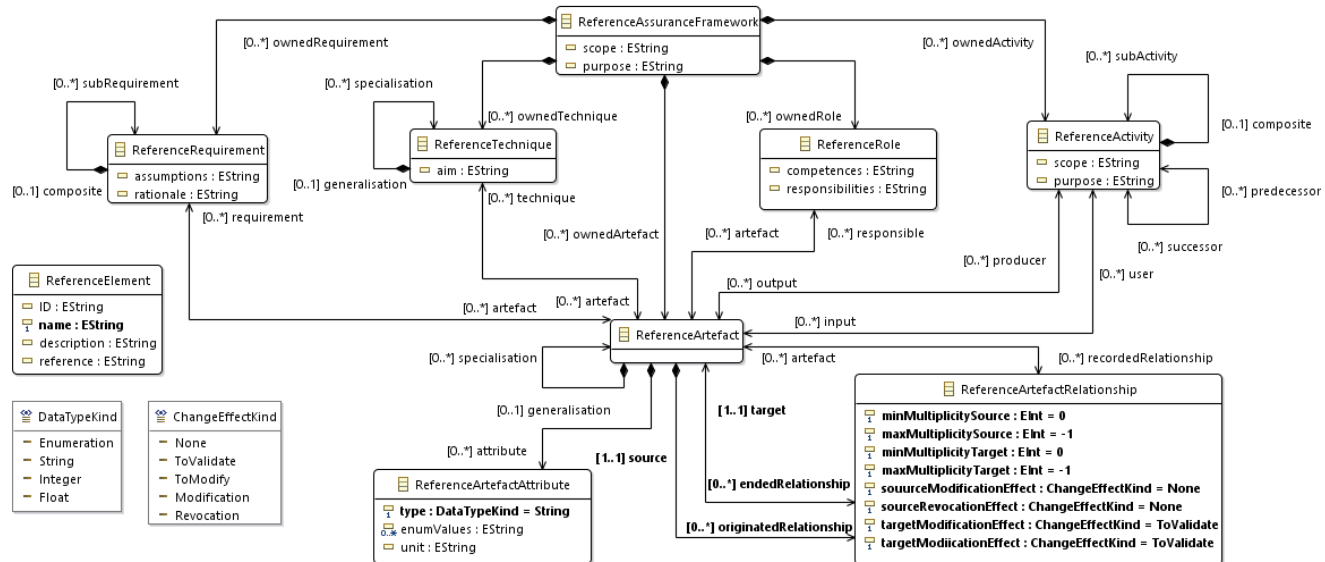
- Lifecycle management of evidence.
- Adaptation to specific standards or characteristics of them.
- Derivation of the structure of information obtained from a standard.
- Monitoring the status of a project.
- Assurance of compliance of the stored information.

To achieve these objectives, six cases have been defined:

- **Use Case 1. Reference Assurance Framework creation:** through an Eclipse plugin, the reference model to be used for a particular project is created. This model will serve to manage compliance with a standard.
- **Use Case 2. Creation of the project:** The user of the application creates a new project, for which he indicates a model created before. Once the user has created a project, he can manage all elements of it.
- **Use Case 3. Monitoring the state:** the application user can review and check at any time what the status of compliance with the standard is.
- **Use Case 4. Manage artifacts:** The user can add, modify and delete artifacts that he may need, according to the model. In addition, the artifacts may be associated with other elements of the project such as techniques, participants, activities, and requirements. The user may also store information about the events and evaluations of the artifacts created.
- **Use Case 5. Manage process information:** The user can add, modify and delete information about techniques, participants and activities, as necessary to demonstrate compliance.
- **Use Case 6. Manage justification information:** the user can add, modify and delete all requirements deemed necessary for compliance.

## Development and results of the plugin

The graphical editor has been created according to this meta-model:



*Ilustración 1: Metamodel for the graphical editor.*

The generation is done from an *.ecore* file, which contains the model definition. From this, a new file is created with *.genmodel* extension, through which the user will be able to begin to create the rest of elements of the editor.

To create the editor, there are several steps to follow:

- **Domain Model:** In the first step for creating the graphic editor, the model *ecore* is added.
- **Domain Gen Model:** once the *ecore* has been added, in this step the user must add the previously generated file *.genmodel*.
- **Tooling Def Model:** at this point a palette is created to create the editor. The figures that will represent each element are added.
- **Graphical Def Model:** in this step the user indicates how the palette is. The nodes, components and connections between them are chosen.
- **Mapping Model:** combining the last three steps, a map representation is created. In this map, all properties of each element are defined. With all this, a new file is created with extension *.gmfmap*

- **Diagram Editor Gen Model:** finally, the last step of the creation is to transform the created map into a diagram editor.

Once the user has done it all, when he runs the project, the appearance of the plugin will be like this:

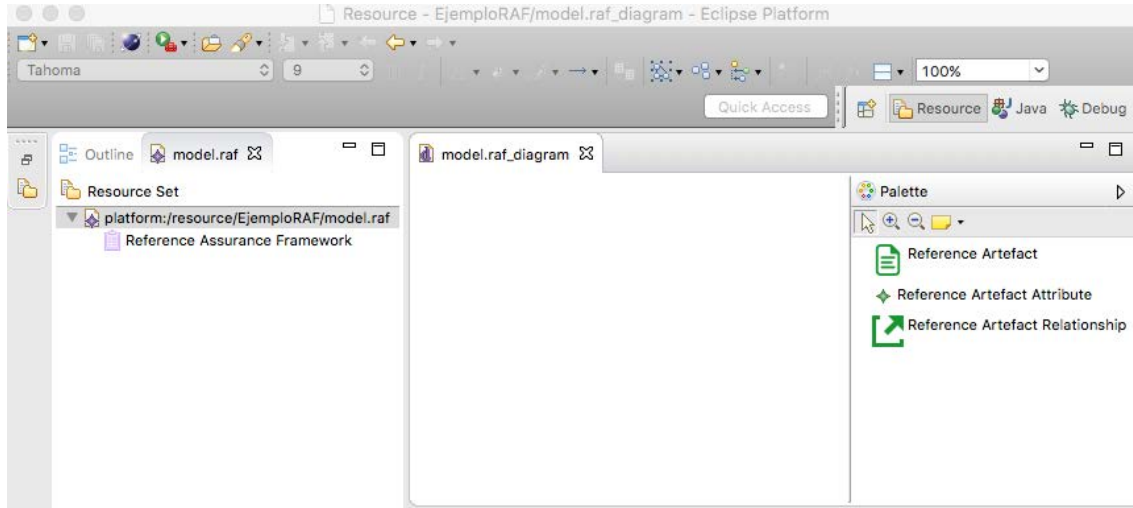


Ilustración 2: Graphical editor appearance.

## Development and results of the application

The methodology that has been carried out to resolve this case is as follows:

- **Stage 1. Planning and description of requirements:** At this stage of the cycle the plan, requirements and use cases to be implemented, the conceptual model and the system architecture are defined.
- **Stage 2. Construction:** Analysis and design of the tool, implementation thereof and tests carried out in the following order:
  - Analysis: Definition of specific requirements, explanation of the use cases.
  - Design: Diagrams related to the analysis phase and relationships between the various elements of the project are detailed. Refinement of the above explained. Schema database definition to be used.
  - Tool coding: Given specified above.
  - Testing: to corroborate the behavior of the application.
- **Stage 3. Installation:** operation of the application developed.

Storing the information entered to the application is carried out through a relational SQL data type. The schema of that database is the following:

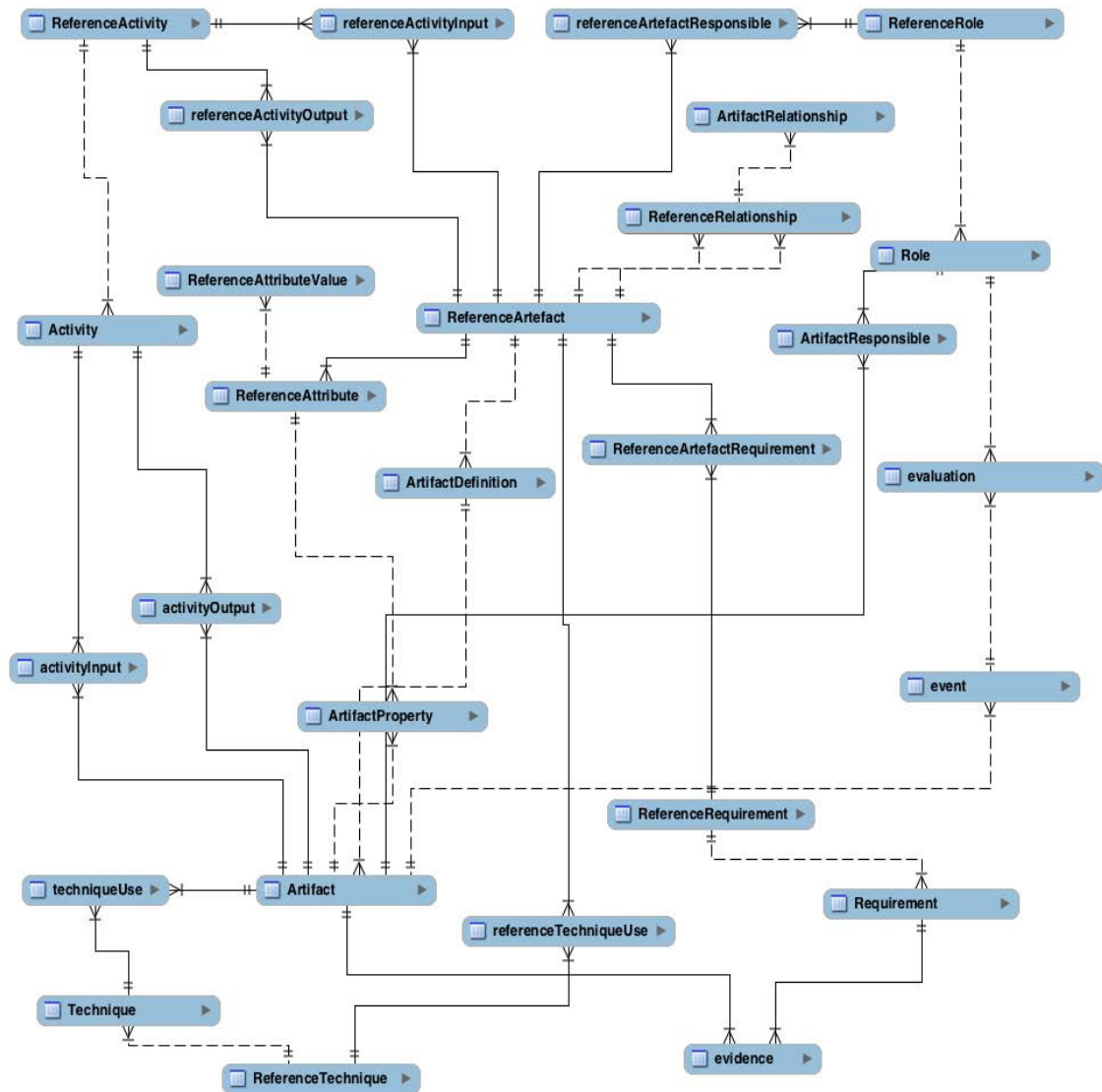


Ilustración 3: Database schema.

The application has been developed following the Model-View-Controller architecture. This architecture implies a separation between the main layers of a software project:

- **Model:** Layer architecture where the user works with data, containing the necessary mechanisms to access and modify the information, it means, all access to database necessary for the proper operation of the application.
- **View:** Layer architecture that facilitates the interaction with the user. It contains the code that allows the viewing of the tool.

- **Controller:** Layer architecture that serves as a link between the view and the model. It contains the necessary elements to carry out the actions and requests of the application code.

With the architecture explained, the project organization is as follows:

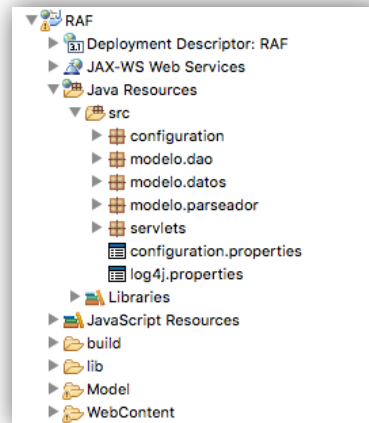


Ilustración 4: Application organization.

- **WebContent:** Folder where the part of the "View" application. Contains all jsps, css and JavaScript using the tool.
- **Model:** Folder where the models that are inserted in the application are automatically stored.
- **Lib:** It contains all the elements necessary for the proper functioning of the tool libraries.
- **Src/configuration:** Part of the application to set the parameters of use.
- **Src/modelo.dao:** Code part that contains all access to database. It contains functions that will allow the application to modify, add or retrieve data. This folder corresponds to the "model" of the architecture.
- **Src/modelo.datos:** Code part that contains the structures and attributes of each of the elements taken into account in the application.
- **Src/modelo.parseador:** It contains part of the code that parses and organizes the contents of the file that is uploaded to the application as a reference model.
- **Servlets:** Contains all the necessary servlets to communicate the application interface to access data, it means, the "view" of the architecture with the "model". It is, therefore, the "controller" of the architecture that is being used.

- **Configuration.properties:** configuration file in which the user will have to enter private data necessary for connection to the database and load the model.
- **Log4j.properties:** log configuration file of the application. The user must enter the path where he wants the file to be generated.

The user communicates with the application through the contents in the "WebContent" folder. From there, depending on the desired action the user calls to different servlets contained in the "Servlets" folder. From these, as needed, communication with different files from other folders ("Src / modelo.dao", "Src / modelo.datos", "Src / modelo.parseador") is performed.

## Conclusions

From the objective point of view, we have achieved the goals set at the beginning of the project, which are satisfied with the work done. These objectives were:

- **Create a graphic model editor**, by Eclipse GMF tool, allowing the user to create a new Reference Assurance Framework for a specific standard whenever the user needs it for a new project.
- **Implementation of a tool that allows a user to add evidence as set out in the model created**, through the application designed to manage evidence, which can manage the different elements that form the model (such as artifacts, requirements, techniques, roles and activities).
- **Monitoring the progress of a project that the user is performing**, through the application that has been developed, in which the user can check at any time the completeness or progress of each reference element.
- **Life cycle management of evidences**, through the deployed application , adding the different versions of each definition artifacts , indicating which is the previous version of the new artifact created , and adding events to control them.

The project undertaken is a first implementation of a possible future project that may cover a number of features that were not taken into account for this development. These are:

- Reporting in the form of documents: Exporting the data entered by the user during the development of the project that is being carried out in the form of an



electronic document, so that all the evidence data collected is stored in a single file.

- Improve the interface: New ways of representation of the information through screen.
- Multi-diagram editor for *Reference Assurance Framework*, with graphics also for other model elements publishers, since in the case that has developed only graphical editing of artifacts and their relationships and attributes are allowed.
- Support for more activities of the evidence life cycle, with features such as analyzing the impact of changes, automating traceability management, visualization of such traceability models or matrices, import information from other tools that create or manage specific artifacts, as for example tools for requirements specification, design, modeling, and testing.

From a personal point of view, this project has served, on the one hand, to learn about software development through models and understand the environment in which these tools are developed. On the other hand, the fact of having to perform a complete and from zero application, and through all phases of development, has served to complete the training received during degree in the different courses taken, and learn to combine the aspects learned in each. The realization of such a project from scratch not only makes the knowledge acquired during the race are completed, but also learn to grow as a worker, to manage time with that account and to acquire knowledge self-sufficiently.

### Total cost of the development

The total project cost is the sum of the costs of personnel and material.

Description	Amount
Personnel costs	5360,00
Material costs	140,91
<b>TOTAL</b>	<b>5500,91</b>

Tabla 2: Total cost of the project.

# 1. Introducción

En este primer apartado se va a explicar el planteamiento del problema inicial y los objetivos que se pretenden conseguir con su resolución, así como la forma de hacerlo. Se detallará a su vez el orden que sigue el documento, y las palabras clave necesarias para su correcto entendimiento.

## 1.1 Motivación

La principal motivación de llevar a cabo este proyecto viene dada por la utilización de nuevas herramientas que no se han visto durante los años de estudios, y que son interesantes a la hora de desarrollar proyectos software.

Un sistema crítico es aquel cuyo fallo puede provocar daños bastante graves, bien a las personas, entorno o economía. Estos sistemas críticos en industrias como la aviación, el ferrocarril o la automoción tienen que cumplir con rigurosos controles de seguridad (*safety*). El objetivo de la certificación de la seguridad es proporcionar un respaldo formal de que el sistema concreto que se quiere asegurar funcionará de manera segura ante los posibles peligros que puedan surgir. Esta certificación se puede asociar con la evaluación bien de productos, procesos o incluso personal.

La seguridad de un sistema se evalúa y garantiza mediante el cumplimiento de los objetivos de seguridad que reducen o mitigan los potenciales riesgos de la seguridad a lo largo de su ciclo de vida del sistema. Estos objetivos de seguridad se suelen establecer a través de un conjunto de criterios reconocidos y aceptados por la industria en la que se desarrollan, conocidos como estándares.

Cada vez más las distintas industrias se rigen por el cumplimiento de estándares y utilizan herramientas de apoyo para ello. Este tipo de herramientas facilitan el almacenamiento de las evidencias que muestran que dichos estándares se cumplen, de forma que los usuarios puedan asegurarlo en el caso de posibles problemas en dichos sistemas críticos.

## 1.2 Objetivos

El objetivo principal de este trabajo es la creación de una herramienta que actúe como editor y repositorio de evidencias de estándares de seguridad que contenga aspectos con los que no cuentan otras herramientas que actualmente están siendo usadas en distintas industrias, como se detallará en puntos siguientes.

Dentro de este objetivo general se pueden diferenciar los siguientes objetivos específicos:

- Creación de un editor gráfico de modelos de estándares.
- Implementación de una herramienta que permita añadir evidencias según lo establecido en el modelo creado.
- Monitorización del progreso del proyecto que el usuario esté llevando a cabo.
- Gestión del ciclo de vida de las evidencias.

## 1.3 Método de Resolución

Para llevar a cabo este proyecto se ha llevado una metodología muy similar a la estipulada por *Craig Larman*. En ella, el proyecto sigue el siguiente curso iterativo e incremental, en forma de cascada, dividido en tres macro-etapas:

- **Planificación y descripción de requisitos:** En esta etapa del ciclo se define el plan a seguir, los requisitos y casos de uso que se van a implementar, el modelo conceptual y la arquitectura del sistema.
- **Construcción:** Análisis y diseño de la herramienta, implementación de la misma y pruebas llevadas a cabo, siguiendo el siguiente orden:
  - Análisis: Definición y explicación de requisitos específicos y casos de uso.
  - Diseño: Diagramas relacionados con la fase de análisis y en los que se detalla el curso y relación de los distintos elementos del proyecto. Refinamiento de lo anteriormente explicado. Definición del esquema de base de datos que se utilizará.
  - Codificación de la herramienta: Teniendo en cuenta lo especificado.
  - Pruebas: para corroborar el funcionamiento de la aplicación.
- **Instalación:** puesta en funcionamiento de la aplicación desarrollada.

## 1.4 Organización del Documento

El documento que se presenta está dividido en los siguientes apartados o epígrafes, cada uno de los cuales contiene la información que se detalla en la siguiente explicación:

- **Capítulo 1. Introducción:** Este capítulo pone en situación al lector, dándole la información necesaria acerca del problema y la metodología usada para su resolución. Además, incluye la estructura del documento y una serie de palabras clave y acrónimos necesarios para conseguir su completa comprensión.
- **Capítulo 2. Análisis de Situación:** El segundo capítulo da una visión más detallada del problema a resolver, situándolo en su contexto, y explicando las causas que promovieron a su resolución y cómo se pretende llegar hasta ésta.
- **Capítulo 3. Análisis del Proyecto:** Sobre el análisis del problema, creando en primer lugar los casos de uso que abarcan los problemas a resolver con la aplicación. Una vez hecho, se expondrá una lista de requisitos que se deberán cumplir para el correcto funcionamiento de la misma.
- **Capítulo 4. Diseño e implementación:** Detalla el diseño realizado a partir de los requisitos del capítulo anterior. Se analizan las distintas alternativas de diseño y arquitectura elegidas. Se exponen, además, una serie de bocetos de la interfaz gráfica de la aplicación. Se explica la forma en la que se ha implementado la aplicación.
- **Capítulo 5. Uso de la herramienta:** Se explican los pasos previos de instalación antes de usar la aplicación y una ruta de uso de la misma, explicando las distintas pantallas a las que se accede y su funcionamiento.
- **Capítulo 6. Pruebas de sistema:** Define las pruebas que se utilizarán para validar el funcionamiento del software. Cada prueba consta de una serie de pasos a seguir para realizarla y los resultados que se espera obtener de ella.
- **Capítulo 7. Gestión del Proyecto:** Se expone la planificación del proyecto, su duración y las fases en las que se divide.

- **Capítulo 8. Conclusiones y Líneas Futuras:** El último capítulo incluye conclusiones acerca del trabajo realizado, y las posibles líneas futuras de la aplicación.
- **Capítulo 9. Bibliografía:** Fuentes utilizadas para completar el documento en el formato de la APA.

## 1.5 Definiciones, Acrónimos y Abreviaturas

En este apartado se definirán aquellas palabras que aparezcan en el documento y que puedan llevar a confusión o no entendimiento del mismo. A su vez, se expondrá una lista de acrónimos y abreviaturas utilizadas durante la realización del mismo.

### 1.5.1 Definiciones

- **Servlet:** Clase, en el lenguaje de programación, utilizada para ampliar las funcionalidades que tiene un servidor web a los que se accede según el modelo de programación *request-response*. (Servlet-UC3M).
- **Sistema crítico de seguridad:** aquel en el que cualquier fallo puede provocar daños irreparables a las personas, propiedad o medio ambiente.
- **Sistema embebido:** sistema de computación que se ha diseñado para llevar a cabo unas funciones específicas, por lo general en sistemas de computación en tiempo real.
- **Modelo:** representación parcial de la realidad que se puede aplicar a cualquier elemento o situación y que cambia en función del objetivo específico que se pretenda conseguir.
- **Evidencia:** Certeza de aquello que no se puede dudar. (RAE, 2016). Consiste en una colección de documentos u objetos que proporcionan las pruebas o apoyos necesarios para corroborar un hecho.
- **Artefacto:** Producto tangible del proceso de desarrollo del software, tal como pueden ser casos de uso o diagramas de clases.

- **Actividad:** unidad de trabajo que requiere, modifica o produce artefactos. Corresponde a una acción que se lleva a cabo en el ciclo de vida del sistema.
- **Requisito:** proposiciones formuladas en relación con la seguridad del sistema, u otro tipo de propiedad del mismo. Condiciones a cumplir.
- **Técnica:** procedimiento específico a través del cual se logra la creación de un artefacto.
- **Rol/Participante:** partes o personas involucradas en la gestión de un artefacto determinado.
- **Reference Artifact:** artefactos específicos de un modelo de un estándar de seguridad determinado.
- **Reference Requirement:** requisitos específicos de un modelo de un estándar de seguridad determinado.
- **Reference Activity:** actividades específicas de un modelo de un estándar de seguridad determinado.
- **Reference Role:** roles o participantes específicos de un modelo de un estándar de seguridad determinado.
- **Reference Technique:** técnicas específicas de un modelo de un estándar de seguridad determinado.

### 1.5.2 Acrónimos y abreviaturas

- MVC: Modelo vista controlador.

## 1.6 Aclaraciones sobre el documento

La normativa para la creación de documentos dice que lo correcto es escribir las palabras en otro idioma entrecomilladas y en cursiva. Debido a la gran cantidad de palabras de habla

inglesa que contiene el documento, puesto a que la implementación de la herramienta se ha hecho en inglés, y para que su lectura sea lo más fácil posible, las palabras escritas en inglés van a escribirse únicamente en cursiva, sin entrecomillar, puesto que se ha considerado que una gran cantidad de esos símbolos hace más difícil la lectura del mismo.

## 2. Análisis de situación

En este apartado del documento se analizará el contexto de la aplicación que se va a desarrollar. El orden seguido para hacerlo será el siguiente:

En primer lugar, se hará una introducción al entorno socio-económico en el que se desarrollan este tipo de aplicaciones y al marco regulador de la información contenida en la misma, siguiendo de una introducción a los sistemas críticos de seguridad, la certificación y los estándares para los mismos. Una vez explicado lo anterior, se pasará a describir cómo funciona la gestión de las evidencias de seguridad, para continuar explicando lo que es la ingeniería dirigida por modelos y presentando modelos de aseguramiento y certificación. Por último, se analizarán algunas de las herramientas relacionadas o similares al proyecto que se ha desarrollado.

### 2.1 Entorno Socio-Económico

Durante los últimos años, las aplicaciones inteligentes y la conexión mediante red se han hecho más importantes e indispensables en el día a día, tanto de los usuarios de forma privada como en las industrias de sistemas integrados, como es el caso del transporte, energía, o incluso salud.

Con todo esto, y cada vez más, existe un problema por garantizar la seguridad, disponibilidad e integridad, entre otras características, de dichos sistemas. Además, debido a la informatización de esos sistemas, se deben respetar ciertas normas de garantía y certificación. Un claro ejemplo de estos casos se da en la aviación. En el caso de los aviones de pasajeros B787 de Boeing, el proceso de certificación tiene una duración de ocho años, con un total de más de 1500 reglamentos que tienen que ser cumplidos para su puesta en funcionamiento. Aunque si es cierto que en la industria de la aviación estos controles son muy estrictos por la importancia de su seguridad, el control en otro tipo de industrias, como la ferroviaria o la automovilística, son similares.

La necesidad de elaborar requisitos de fiabilidad y de cumplir normas de garantía hace que el desarrollo de los sistemas críticos sea mucho más lento y costoso que otros tipos de sistemas. Por ejemplo, un software no crítico embebido puede suponer un coste total de



aproximadamente entre 15 y 40 dólares por línea de código, mientras que el de un software critico puede llegar a la cantidad de 1000 dólares (*Koopman, 2010*).

Para demostrar el cumplimiento de las normas o reglamentos que se establecen para estos sistemas, se deben llevar a cabo una serie de pruebas, que han ido evolucionando y haciéndose más robustas con el paso del tiempo. Un ejemplo de esto se da en el sector de la sanidad. Los productos sanitarios, desde un principio, tienen que ser vistos como no fiables, hasta que todas las pruebas que se realicen sobre ellos digan lo contrario. Solo en ese caso se demuestra su fiabilidad y correcto funcionamiento. Los estudios de fallos en este tipo de sistemas han identificado carencias importantes en el aseguramiento de las pruebas, que tuvieron que ver con deficiencias en los requisitos establecidos en un primer momento. Uno de los problemas principales que se dieron fue la falta de detalle en dichas pruebas de aseguramiento, como por ejemplo en la definición de sus características. En dos estudios llevados a cabo recientemente, profesionales del sector de la medicina destacaron que las técnicas de valoración de dichas pruebas, a día de hoy, no se usan demasiado, y que existen detalles sobre cambios en ellas que pueden afectar a otras y que no se registran. Esta falta de precisión en dichas pruebas de aseguramiento concluye que aún no existen garantías bien definidas en este sector para hacer frente a los riesgos que puedan darse.

Europa representa a día de hoy el 30% de la producción mundial de sistemas embebidos, con ventaja sobre otras regiones en aplicaciones industriales, de automoción, médico y aeroespacial. El mercado mundial de este tipo de sistemas se estima en 850 millones de euros. Dentro de estas cifras, el sector aeroespacial, automovilístico y medico representan en su conjunto un total de 472 millones en el mundo, siendo la cifra europea 141 millones de euros. Dentro del ámbito laboral, este tipo de desarrollo de sistemas generan cada año, aproximadamente, 50.000 puestos de trabajo.

## 2.2 Marco regulador

Existen dos normas principales a tener en cuenta que hacen referencia al tipo de datos e industria en la que la aplicación se desarrolla o distribuye. Estas son: La Ley Oficial de Protección de Datos y la Ley de Propiedad Industrial.

- **Ley Oficial de Protección de Datos (LOPD):** tiene como objetivo la regulación del tratamiento de los archivos y datos de carácter personal, sea cual fuere el soporte en

el que se traten, y los derechos y obligaciones que los ciudadanos tienen sobre ellos (Boletín Oficial del Estado, 2015). Los usuarios del proyecto deberán introducir datos personales tales como el email. El cumplimiento de esta ley garantizará la no distribución de los mismos.

- **Ley de Propiedad Industrial:** Conjunto de derechos que pueden tener las personas, bien sean físicas o jurídicas, sobre una invención, una marca o un diseño industrial. Da a la persona el derecho de utilizarla y el derecho a prohibir que lo haga una tercera persona (Boletín Oficial del Estado, 2015). La aplicación llevará la marca del desarrollador y sólo podrá ser utilizada por sus clientes.

A parte de las normativas generales que se tienen que cumplir en el país, existen normas específicas que también hay que tener en cuenta, que no afectan directamente a la aplicación que se desarrolle, pero sí a la industria en la que se usa o implanta, ya que la certificación de los sistemas críticos es un requisito legal en varios dominios de aplicación. En el caso de Europa, estas son:

- **Sector ferroviario:** la ERA (*European Railway Agency*) es la encargada de la publicación de las normas. Este sector se rige por la familia de estándares CENELEC (ERA, 2016).
- **Sector aeroespacial:** la ESA (*European Space Agency*) es la encargada de la publicación de las normativas a seguir. Este sector se rige por la familia de estándares ECSS (ESA, 2016).
- **Sector de la aviación:** la EASA (*European Aviation Safety Agency*) publica las normativas relacionadas con este sector. Se rige por la familia de estándares DO (como por ejemplo el estándar DO-178) y ARP (EASA, 2016).
- **Dispositivos médicos:** en este caso, cada país tiene normas o certificaciones independientes, aunque todas tengan puntos en común. Existe también normativa a nivel europeo que deben cumplir, marcada por la CE (CE, 2016).
- **Automoción:** a día de hoy la certificación no es un requisito legal, pero se suele evaluar y verificar el cumplimiento de la norma ISO-26262.

- **Otros dominios** como los sistemas industriales o la energía nuclear y gas también están regulados por normativas específicas.

## 2.3 Sistemas críticos de seguridad

Se conoce como “Sistema crítico de seguridad” aquel en el que cualquier fallo puede provocar daños irreparables a las personas, propiedad o medio ambiente (*Nair, 2014*).

Algunos ejemplos de estos sistemas se ven continuamente en la vida cotidiana. Estos son, por ejemplo, los sistemas de control de trenes, metros o coches, los sistemas de gestión de tráfico aéreo, las señales de tráfico o los sistemas de monitorización sanitarios. Por lo general, y cada vez más, estos sistemas dependen de equipos informáticos para su correcto funcionamiento (*Íbidem*).

Los sistemas de seguridad críticos se suelen usar para controlar procesos que ya se han definido anteriormente. El sistema realiza un seguimiento de los procesos, normalmente, mediante sensores. Dichos sensores envían información al software encargado del proceso, de forma que, en el caso que sea necesario, tome medidas para contrarrestar el fallo que se esté produciendo. El software en cuestión enviará la nueva información al equipo a través de algún tipo de actuador, que se encargará de cambiar los parámetros necesarios (*Íbidem*).

Estos sistemas, a su vez, suelen tener un componente humano, encargado de controlar y supervisar el funcionamiento del mismo (*Íbidem*).

## 2.4 Certificación de sistemas críticos de seguridad

La certificación de la seguridad tiene como objetivo garantizar que un sistema se ha considerado seguro para un determinado uso, mediante el establecimiento de objetivos de seguridad en el entorno en el que operará el sistema (*Nair, 2014*).

Los sistemas de seguridad críticos, por lo general, están sujetos a procesos de certificación de seguridad a través de licencias y organismos reguladores. La seguridad de un sistema es un punto fundamental a la hora de desarrollarlo, y debe tenerse muy en cuenta en las primeras fases de su creación. Los objetivos de seguridad que se establecerán deberán

estar basados en los peligros potenciales a los que se puede enfrentar el sistema que se esté desarrollando (*Íbidem*).

La evaluación de la seguridad trae consigo una serie de actividades de análisis y evaluación relacionados con los objetivos previamente fijados, y que se llevan a cabo durante todo el ciclo de vida del sistema concreto. Los objetivos de seguridad se establecen según unas normas establecidas en cada tipo de industria, conocidas como estándares, que se explicarán en el siguiente apartado de éste epígrafe.

Para demostrar que se satisfacen los objetivos de seguridad marcados, se lleva a cabo una recopilación de evidencias durante el ciclo de vida del sistema. Dichas evidencias se presentan, una vez realizadas, a una entidad de certificación de terceros, que evaluará el sistema y, en caso de que todo este correctamente, emitirá un certificado, permitiendo el despliegue y la puesta en marcha del sistema.

## 2.5 Estándares para sistemas críticos de seguridad

Se entiende por norma o estándar al conjunto de requisitos que debe cumplir un determinado sistema, independientemente del sector en la que se desarrolle (*Nair, 2014*).

Las normas de certificación que se utilizan para el proceso de certificación pueden ser nacionales o internacionales, y genéricas (aplicadas a un sector específico) o específicas (aplicadas a un tipo de sistema concreto dentro de ese sector) (*Íbidem*).

Algunos ejemplos de estándares de seguridad para sistemas críticos son:

- DO-178C, utilizado expresamente para la certificación del software destinado a los sistemas o equipos de a bordo (aviación).
- IEC61508, utilizado para la evaluación de certificación de sistemas relacionados con la seguridad en el ámbito electrónico o programable. Se trata de unos de los estándares más utilizados.

Por lo general, un sistema específico puede ajustarse a varios estándares, y dependerá del cliente o del contrato que se tenga con él cuál es el que se va a seguir. El uso o la certificación del sistema por parte de un estándar da al usuario la seguridad de que su

sistema ha sido probado y certificado por un órgano regulador. Este certificado se puede usar también en caso de fallos, dejando constancia de que, aunque se haya producido un error, el sistema cumplía con la normativa reguladora del mismo (*Íbidem*).

## 2.6 Gestión de evidencias de seguridad

Las evidencias de seguridad son la forma de demostrar que se han probado y cumplido los objetivos especificados para un proyecto determinado. Son un elemento muy importante de la certificación de la seguridad de los sistemas críticos, ya que son la demostración del cumplimiento de la norma.

Una evidencia consiste, de forma básica, en una colección de documentos u objetos que proporcionan las pruebas o apoyos necesarios a una petición. Puede darse el caso de que se hagan afirmaciones o nuevas evidencias sobre evidencias anteriores, de forma que se re-pruebe o confirme una afirmación hecha.

Cuando se realiza la gestión de evidencias en un proyecto, el primer paso es determinar cuáles son las pruebas o evidencias que se deben proporcionar. Una vez hecho esto, se deben recoger y podrán ser evaluadas o añadidas a una cadena de evidencias. Durante el proceso, puede darse el caso de que haya que cambiar algunos elementos de las pruebas que se han definido y que, estos cambios, afecten a otros elementos. Por esto, pueden aparecer problemas, como por ejemplo incoherencia entre distintas pruebas o elementos. Una vez que el conjunto de las pruebas del proyecto se haya considerado como adecuado, el proceso puede darse por terminado (OPENCROSS).

A continuación se muestra un esquema del ciclo de vida común de las evidencias:

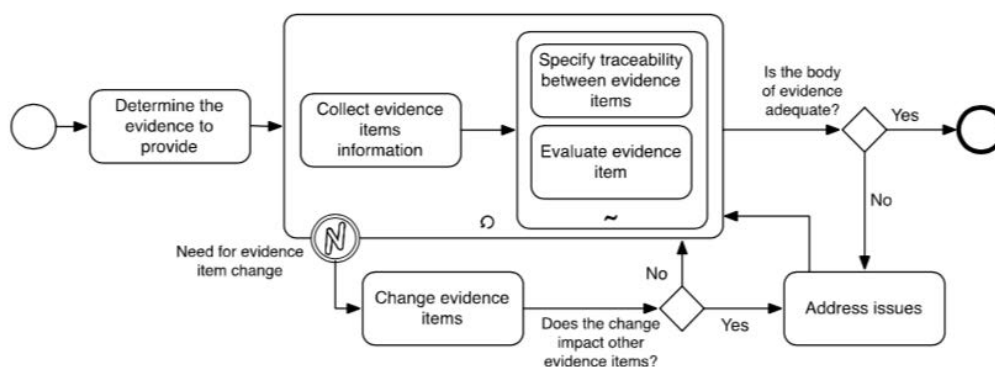


Ilustración 5: Modelo del proceso de la certificación de las evidencias.

## 2.7 Ingeniería dirigida por modelos

Para explicar lo que implica la ingeniería dirigida por modelos hay que, en primer lugar, definir en qué consiste un modelo.

Se define modelo como “Una representación abstracta y parcial de un sistema o proceso”. A través de esta definición se puede extraer que un modelo es, por tanto, una representación parcial de la realidad que se puede aplicar a cualquier elemento o situación y que cambia en función del objetivo específico que se pretenda conseguir. En el entorno del software, la utilización de modelos permite al usuario almacenar información de forma estructurada, facilitando su comprensión (*Nair, 2014*).

Los modelos ayudan a comprender un problema complejo y sus posibles soluciones a través de la abstracción de la información. El desarrollo de sistemas software suele estar entre los problemas más complejos de resolver, y puede beneficiarse del uso de modelos y técnicas de modelados. Sin embargo, los modelos en ingeniería del software aún no son muy frecuentes, o todo lo que podrían serlos, y suelen desempeñar un papel secundario en el desarrollo de un sistema. Aun así, los potenciales beneficios de la utilización de modelos en el software son mayores que en cualquier otro tipo de industria (*IEEE Software, 2003*).

La ventaja principal del desarrollo a través de modelos es que los conceptos están más relacionados con el problema a resolver que con la tecnología de implementación.

Cualquier proceso dirigido por modelos deberá tener un objetivo específico a cumplir, y deberá también tener una representación de la realidad a explicar en función de lo que se necesite. Cada modelo se implementa conforme al meta-modelo que lo contiene, tal y como se explicará en el siguiente apartado respecto a la certificación de sistemas críticos de seguridad.

## 2.8 Modelos para la certificación de sistemas críticos de seguridad

A continuación se van a exponer distintos modelos desarrollados para dar apoyo y facilitar la certificación de sistemas críticos de seguridad.

**SACM** es un modelo de evidencias que define una serie de elementos para construir e intercambiar declaraciones relacionadas con pruebas para el aseguramiento de sistemas, lo que facilita el desarrollo de nuevas herramientas de aseguramiento. Proporciona la base para el diseño lógico de herramientas construidas para almacenar, gestionar, evaluar e informar los elementos de prueba. Algunas de las características que pueden tener las evidencias en este modelo son ser directas o indirectas, primarias o secundarias, o de una fuente original o derivada. El modelo de SACM es el siguiente:

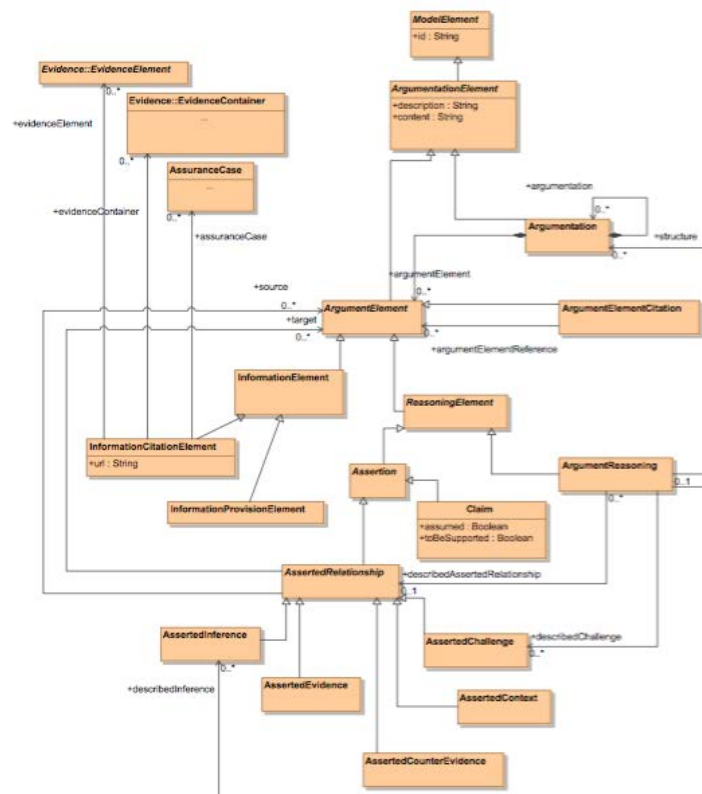
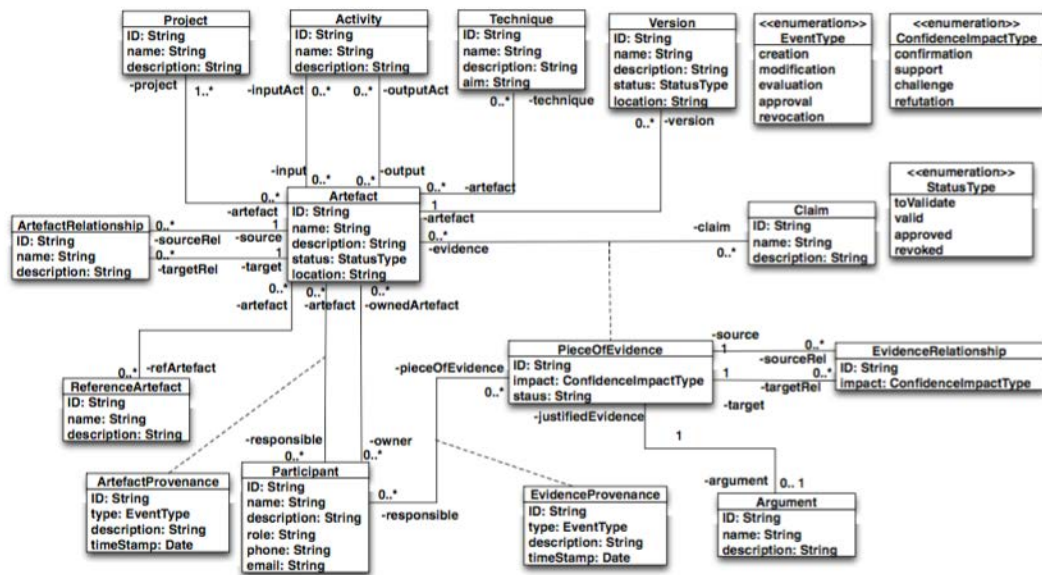


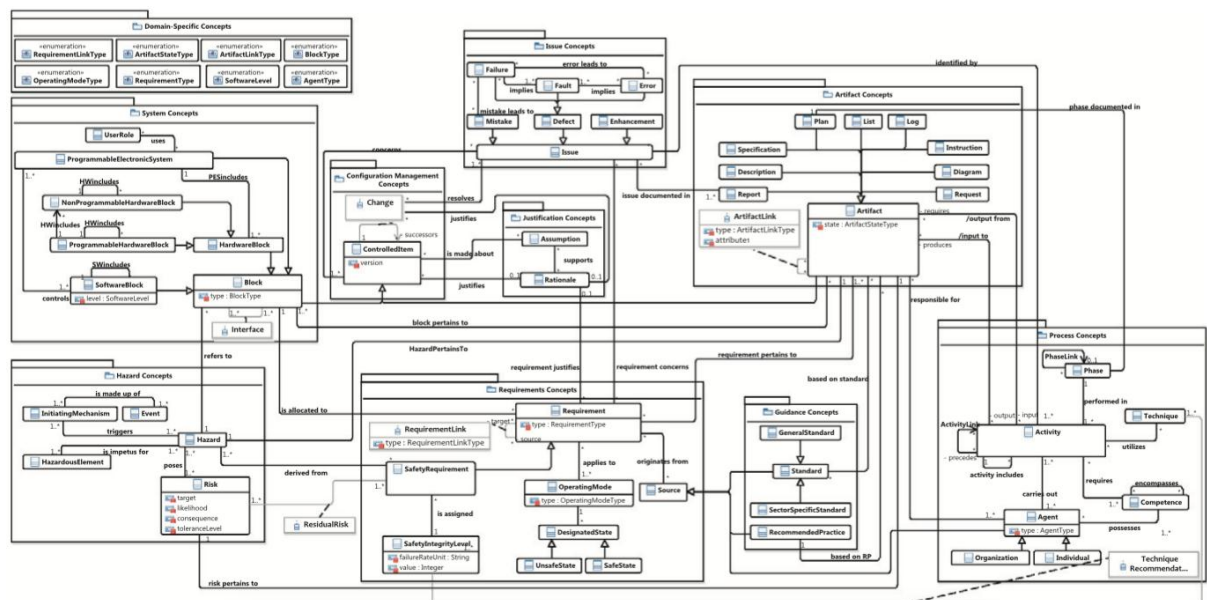
Ilustración 6: Modelo de SACM.

**SafeTIM** proporciona el conjunto de conceptos y relaciones necesarias para gestionar correctamente la trazabilidad de evidencias de seguridad en entornos industriales reales. Además de distinguir entre los objetos que se utilizan durante el ciclo de vida del sistema y su utilización como evidencia, trata de proporcionar una visión global de toda la información asociadas con las evidencias. El modelo conceptual de SafeTIM es el siguiente (Nair, 2014):



*Ilustración 7: Modelo de SafeTIM.*

***The IEC61508 Model***, con un total de diez paquetes, contiene las abstracciones necesarias para el modelado de los principales conceptos de la normativa IEC61508. Su esquema, en formato UML, es el siguiente:

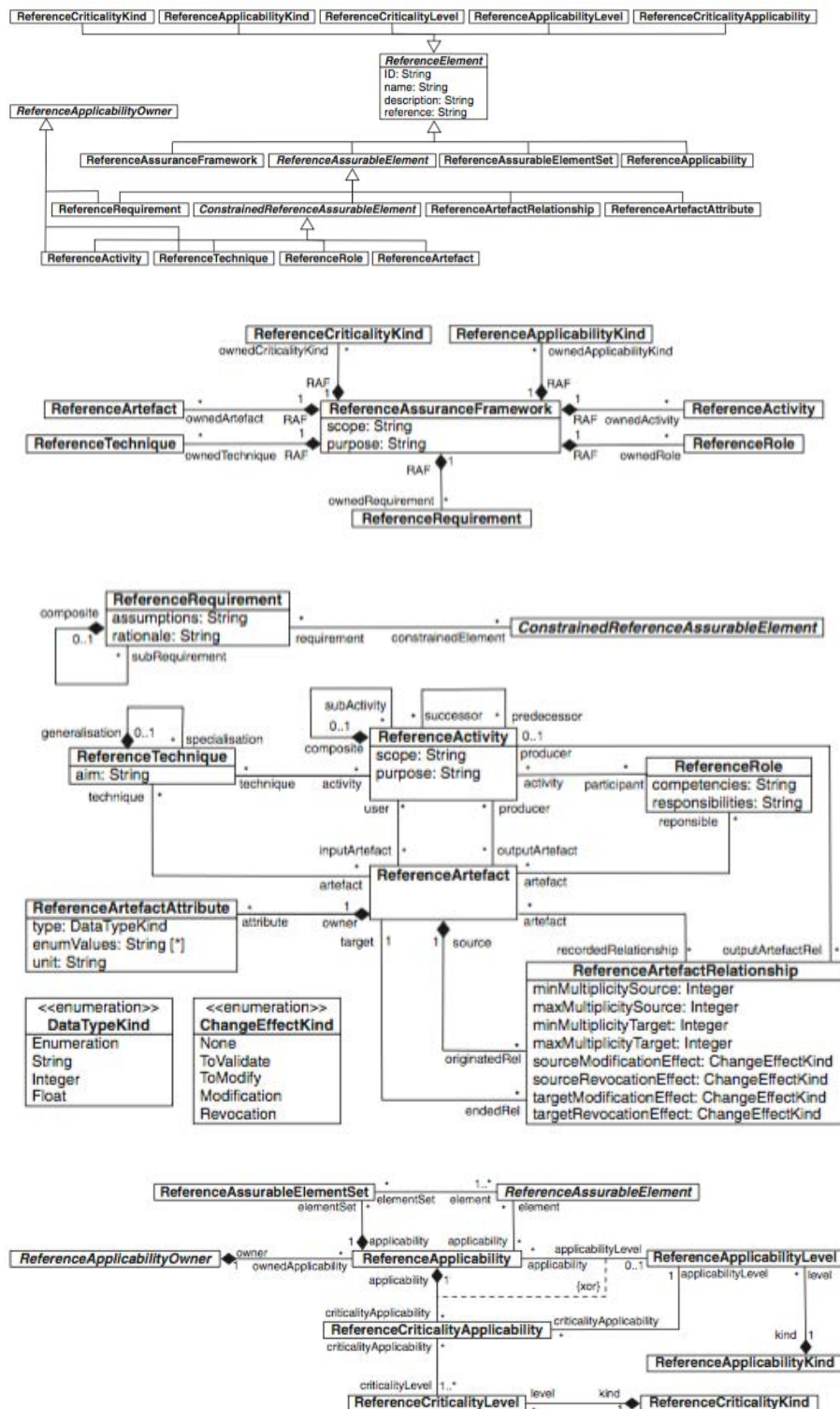


*Ilustración 8: Modelo de IEC61508.*

**Reference Assurance Framework**, cuyo meta-modelo incluye los conceptos claves y las relaciones entre los mismos con el fin de demostrar el cumplimiento de un estándar de



seguridad. Mediante la creación de un modelo basado en el meta-modelo se persigue evitar la subjetividad en la interpretación de cada elemento del estándar. El meta-modelo RAF también proporciona un medio común para comparar las normas de seguridad, a través de una terminología común. El modelo es el siguiente:



*Ilustración 9: Modelo de RAF.*

El meta-modelo utilizado durante este proyecto para la certificación de los estándares es una simplificación del ofrecido por RAF. Dicho meta-modelo es el siguiente:

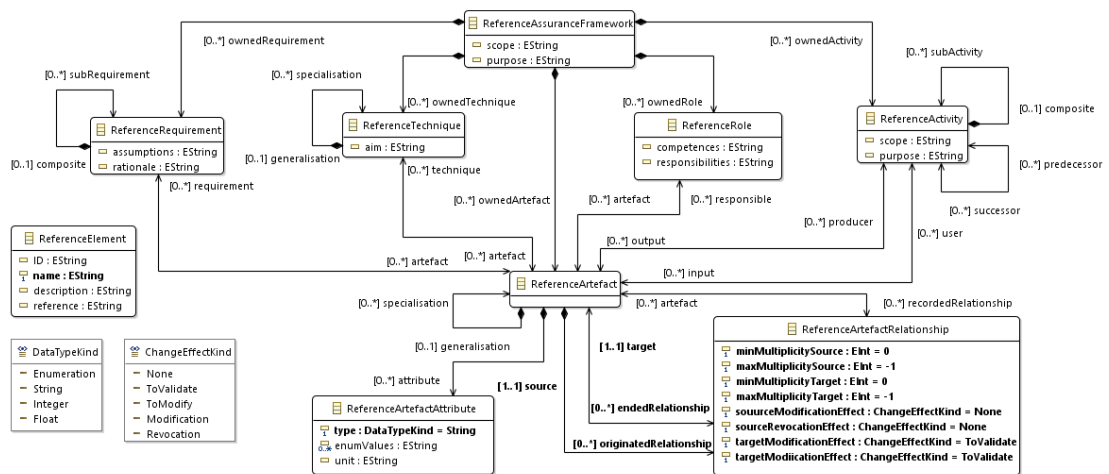


Ilustración 10: Meta-modelo utilizado para el desarrollo del proyecto.

## 2.9 Herramientas para la certificación de sistemas críticos de seguridad

En este punto del documento se exponen algunos ejemplos de herramientas similares a la desarrollada en este proyecto para la gestión de evidencias.

**LDRA** es una herramienta de verificación del estándar ISO 26262, sobre los sistemas de seguridad en automóviles. Está basado en evidencias y funciona de la siguiente forma: Para los objetivos, muestra una lista ordenada de los aspectos a verificar del estándar, y su cumplimiento (LDRA).

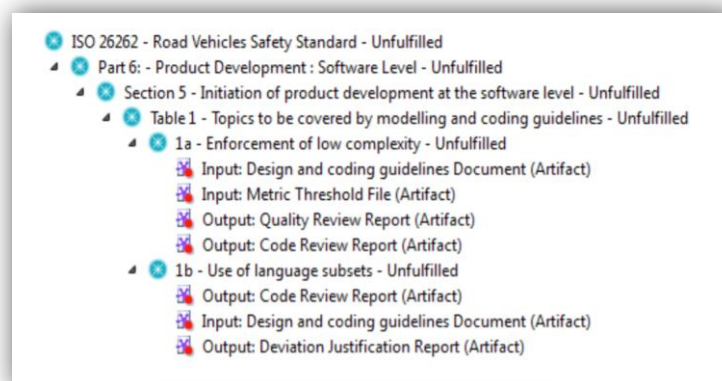


Ilustración 11: Interfaz principal de LRDA. (LDRA)

**GoedelWorks**, de la compañía Altreonic, da apoyo a la verificación de la conformidad con estándares tales como IEC61508, IEC62061, ISO26262, ISO13849, ISO25119, ISO15998, CMMI y “Automotive SPICE”. Esta implementado y disponible bajo el modelo de negocio SaaS (Altreonic). La apariencia de este sistema es la siguiente:

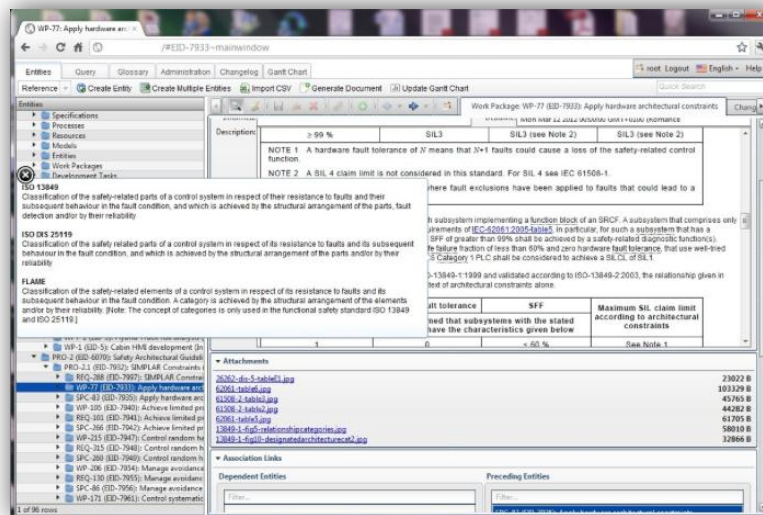


Ilustración 12: Interfaz Principal de GoedelWorks. (Altreonic)

**OPENCROSS**, basado en evidencias, se ha desarrollado en el contexto de un proyecto de investigación europeo a gran escala dedicado a la certificación de la seguridad para los mercados ferroviario, de aviación, y de automoción. Permite la elección del modelo de referencia, es decir, la adaptación a estándares específicos y seguir el ciclo de vida de las evidencias introducidas (OPENCROSS). El aspecto principal de la herramienta es el siguiente:

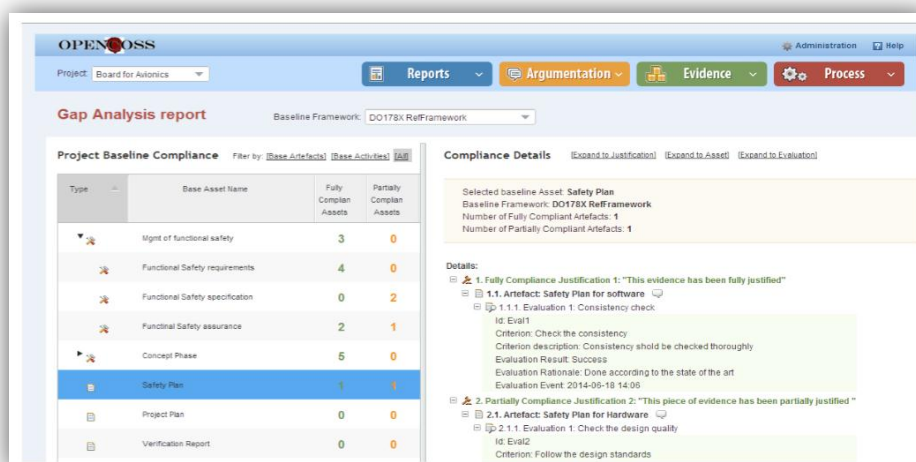


Ilustración 13: Interfaz Principal de OPENCROSS (OPENCROSS)

**CRESCO**, es una herramienta de construcción de repositorios de evidencias. Dibuja un modelo para generar seguidamente un esquema de base de datos según las evidencias añadidas. Muestra a su vez mediante una aplicación web un repositorio para poder manipularlas (CRESCO, 2011).

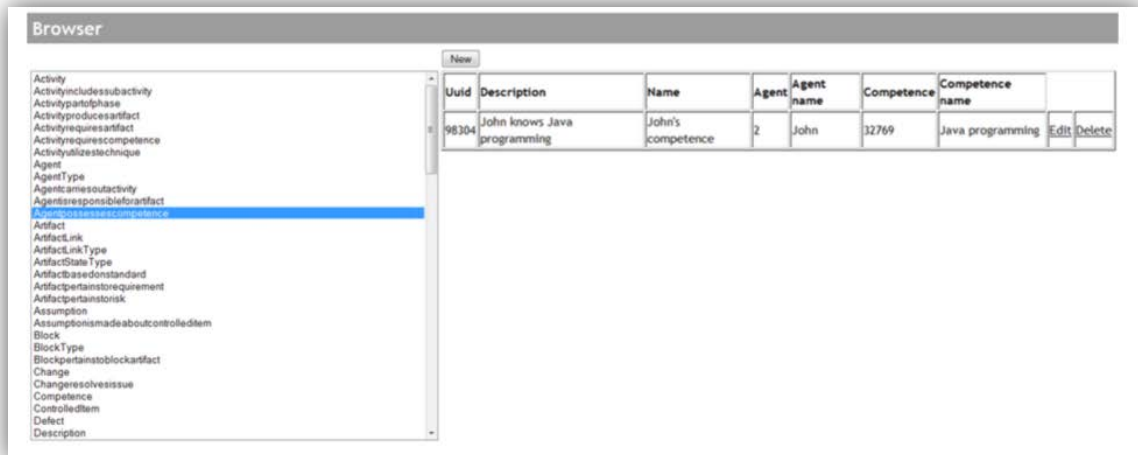


Ilustración 14: Ejemplo de salida de interaz de CRESCO. (CRESCO, 2011)

**WEFACT**, que facilita la validación, verificación y certificación de los sistemas de seguridad de forma modular. Resume o compacta la información de los procesos y da una base de la certificación. Puede integrar herramientas externas (WEFACT).

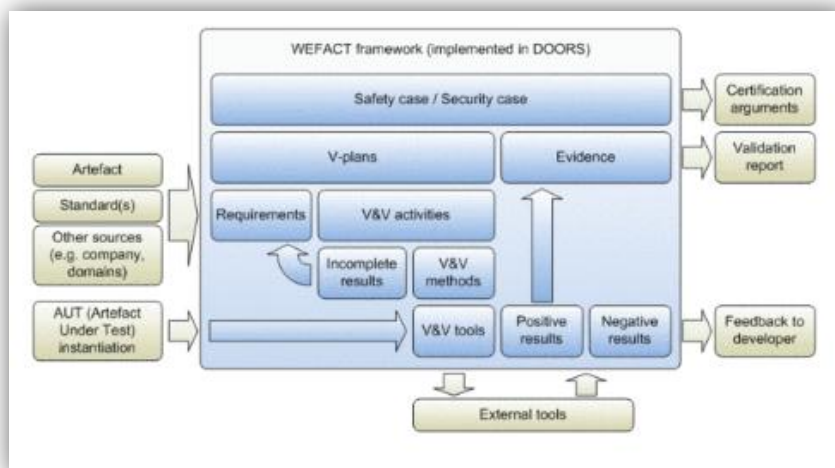


Ilustración 15: Esquema de uso de WEFACT. (WEFACT).

**PTC Integrity Process Director**, proporciona un enfoque práctico para establecer medidas, gestión y mejora de los distintos procesos operativos de una empresa. Se basa en la mejora de los procesos y en proyectos reales (Director).

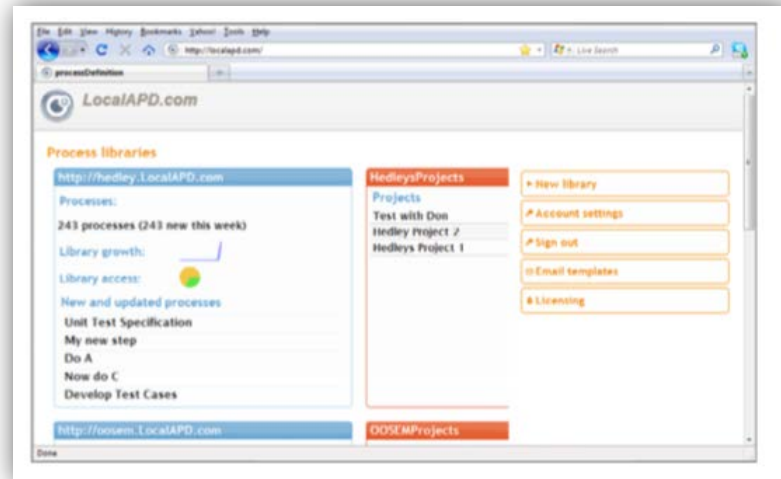


Ilustración 16: Interfaz principal de PTC Integrity Process Director (Director)

**PTC Integrity Process Check**, responde a los cambios en los archivos, webs, y artefactos de gestión de configuración, entre otros. Incluye listas de comprobación para los estándares DO-178 y DO-254 (Check).

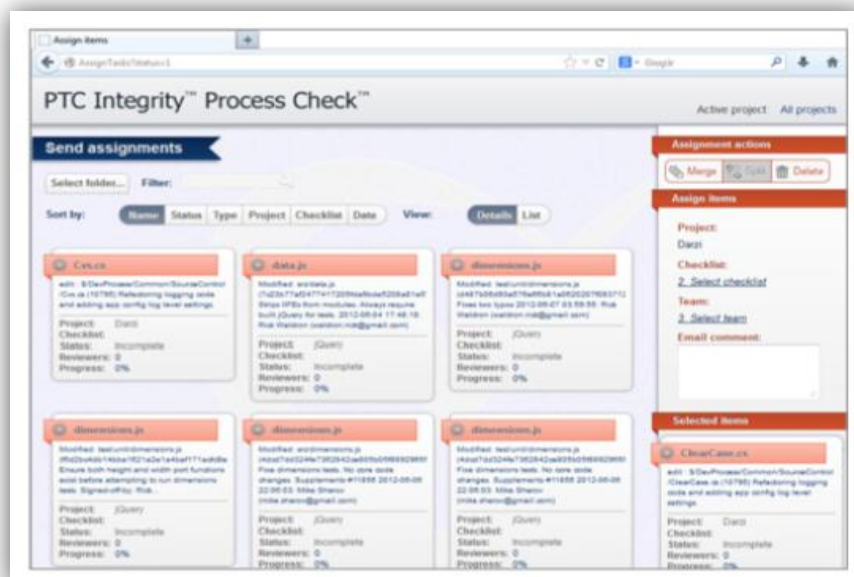


Ilustración 17: PTC Integrity Process Check (Check)

## 2.10 Diferencias entre las aplicaciones existentes y la implementada

La herramienta que se va a desarrollar implementa aspectos a las que no dan apoyo las que se han expuesto. Estos son:

- Gestión del ciclo de vida de las evidencias.
- Adaptación a estándares específicos o características de ellos.
- Derivación de la estructura de la información a partir de la obtenida de un estándar.
- Seguimiento del estado de un proyecto.
- Aseguramiento de la conformidad de la información almacenada.

En el siguiente cuadro se muestran cuáles de las herramientas anteriormente expuestas implementan los puntos anteriores. La forma en la que se va a representar es la siguiente:

- √: cumple con las funcionalidades.
- - : No cumple con la funcionalidad completamente, pero sí parcialmente.
- X: No cumple con la funcionalidad.

	LDRA	GoedelWorks	OPENCOS	CRESCO	WEFACT	PTC Director	PTC Check
Gestión del ciclo de vida	-	-	√	-	-	-	-
Adaptación a estándares	X	X	√	-	X	-	-
Derivación de la estructura	X	X	X	√	X	X	X
Seguimiento del estado	√	X	√	X	X	√	√
Aseguramiento de información	-	-	X	√	√	√	√

Tabla 3: Funcionalidades implementadas por las herramientas similares.



## 3. Análisis del Proyecto

En esta sección se dará una visión de los objetivos que se buscan conseguir, especificándolos casos de uso y requisitos necesarios para cumplirlos, y qué elementos del proyecto se verán afectados por los mismos.

Se detallarán los distintos casos de uso, seguidos de los requisitos, clasificados según su tipo.

### 3.1 Casos de uso

#### 3.1.1 Definición

En el proyecto se pueden distinguir seis casos de uso principales, cada uno con el tipo de usuario correspondiente, como se observa en la siguiente figura:

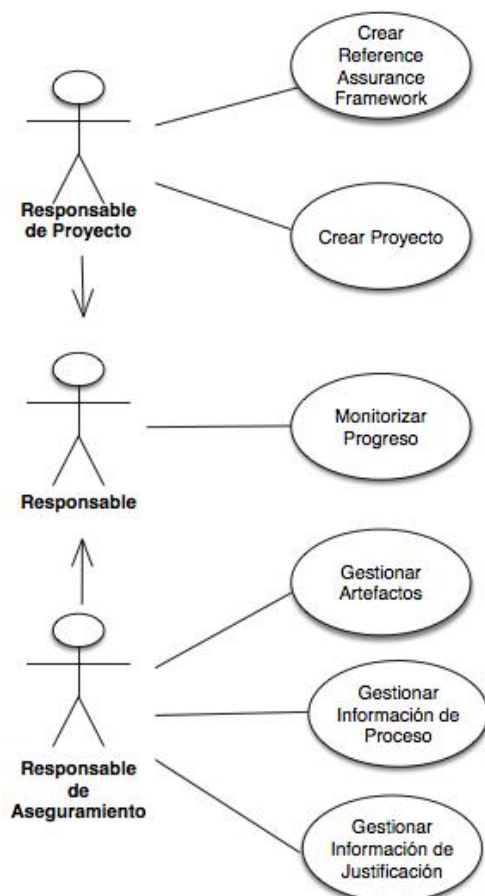


Ilustración 18: Diagrama de Casos de Uso

### 3.1.2 Modo de exposición

El formato que se seguirá para exponer los distintos casos de uso de la herramienta será el siguiente:

<b>Identificador</b>	
<b>Título</b>	
<b>Actor</b>	
<b>Propósito</b>	
<b>Descripción</b>	
<b>Pre-condiciones</b>	
<b>Pasos</b>	
<b>Alternativas</b>	
<b>Post-condiciones</b>	

Tabla 4: Formato de los casos de uso.

- El formato del identificador será: CU-XX, siendo CU las siglas correspondientes a la frase “Caso de Uso”, y XX el número que corresponda al mismo.
- Título: Nombre del caso de uso identificativo de su finalidad.
- Actor: Rol que llevará a cabo las acciones del caso de uso.
- Propósito: Objetivos a cumplir con el caso de uso.
- Descripción: Breve explicación del caso de uso que se está analizando.
- Pre-condiciones: Condiciones que han de darse antes de realizar las acciones que se correspondan con el caso de uso.
- Pasos: Línea a seguir para conseguir el propósito del caso de uso.
- Alternativas: Otras opciones de realización.
- Post-condiciones: Estado en el que debe quedar el sistema una vez ejecutadas las acciones en circunstancias normales de uso.



## 3.1.3 Casos de Uso

<b>Identificador</b>	CU-01
<b>Título</b>	Crear <i>Reference Assurance Framework</i> .
<b>Actor</b>	Responsable de proyecto.
<b>Propósito</b>	Crear el modelo de referencia con el que se va a trabajar a partir de un estándar concreto.
<b>Descripción</b>	El responsable del proyecto podrá crear tantos modelos de referencia como estándares quiera tener en cuenta a la hora de llevar a cabo un proyecto.
<b>Pre-condiciones</b>	<ul style="list-style-type: none"> <li>El plugin o herramienta de creación tiene que estar abierta.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. El responsable del proyecto crea un nuevo archivo tipo "GMF Project".</li> <li>2. El responsable del proyecto crea los artefactos, sus atributos y las relaciones de los mismos que considere necesarios mediante el diagrama.</li> <li>3. El responsable del proyecto crea el resto de elementos necesarios para completar el modelo mediante el árbol, añadiendo los hijos que se consideren necesarios a cada elemento.</li> </ol>
<b>Alternativas</b>	<ol style="list-style-type: none"> <li>1. En caso de que se produzca algún error, éste se muestra por consola.</li> </ol>
<b>Post-condiciones</b>	<ul style="list-style-type: none"> <li>Se ha creado un archivo tipo .raf que se podrá usar para la creación del proyecto.</li> </ul>

Tabla 5: Caso de uso "Crear Reference Assurance Framework"

<b>Identificador</b>	CU-02
<b>Título</b>	Crear proyecto.
<b>Actor</b>	Responsable de proyecto.
<b>Propósito</b>	Creación de un nuevo proyecto.
<b>Descripción</b>	El responsable del proyecto crea un nuevo proyecto sobre el cual se llevarán a cabo el resto de especificaciones y casos de uso.
<b>Pre-condiciones</b>	<ul style="list-style-type: none"> <li>• La aplicación tiene que estar abierta.</li> <li>• Debe haber un modelo que sirva de referencia que el usuario pueda elegir.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. El responsable del proyecto abre la aplicación.</li> <li>2. El responsable del proyecto va a la pantalla de "Configuración".</li> <li>3. El responsable del proyecto introduce el nombre del proyecto, la descripción y selecciona el archivo que contiene el modelo sobre el que se va a trabajar.</li> </ol>
<b>Alternativas</b>	<ol style="list-style-type: none"> <li>1. Si ha habido algún error a la hora de crear el proyecto, se le notificará al responsable del proyecto mediante un mensaje por pantalla.</li> <li>2. El error aparecerá también en el <i>log</i> de la aplicación</li> </ol>
<b>Post-condiciones</b>	<ul style="list-style-type: none"> <li>• Se ha creado el proyecto, y éste aparece en la lista de proyectos en curso.</li> <li>• Al acceder al nuevo proyecto, se verán los elementos de referencia del modelo elegido, y el resto de la aplicación vacía para que se pueda empezar a trabajar sobre él.</li> </ul>

Tabla 6: Caso de uso "Crear Proyecto"

<b>Identificador</b>	CU-03
<b>Título</b>	Monitorizar progreso.
<b>Actor</b>	Responsable.
<b>Propósito</b>	Comprobar el estado de avance del proyecto con el cual se está trabajando.
<b>Descripción</b>	El responsable podrá visualizar en todo momento el estado del proyecto con el que se está trabajando en un momento dado.
<b>Pre-condiciones</b>	<ul style="list-style-type: none"> <li>• La aplicación debe estar lanzada.</li> <li>• Debe haber un proyecto seleccionado.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. El responsable selecciona un proyecto de los que haya creados a través de la configuración de la aplicación.</li> <li>2. El responsable accede a la pantalla "Proyecto".</li> </ol>
<b>Alternativas</b>	<ol style="list-style-type: none"> <li>1. Si se produce un error al acceder a la pantalla se notificará al responsable del fallo.</li> <li>2. El error aparecerá también en el <i>log</i> de la aplicación</li> </ol>
<b>Post-condiciones</b>	<ul style="list-style-type: none"> <li>• El responsable ha visto el avance del proyecto.</li> </ul>

Tabla 7: Caso de uso "Monitorizar Progreso"

<b>Identificador</b>	CU-04
<b>Título</b>	Gestionar artefactos.
<b>Actor</b>	Responsable de aseguramiento.
<b>Propósito</b>	Completar todo lo referido a los artefactos del proyecto.
<b>Descripción</b>	El responsable de aseguramiento deberá añadir los artefactos necesarios para completar de forma satisfactoria el proyecto. Una vez añadidos, podrá modificarlos, asociarlos con otros elementos y borrarlos.
<b>Pre-condiciones</b>	<ul style="list-style-type: none"> <li>• La aplicación debe estar lanzada.</li> <li>• Se debe haber seleccionado un proyecto a través de la configuración de la aplicación.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. El responsable de aseguramiento selecciona un proyecto de la lista de proyectos creados.</li> <li>2. El responsable de aseguramiento selecciona uno de los <i>Reference Artifact</i> disponibles para añadir una nueva</li> </ol>

	<p>definición.</p> <ol style="list-style-type: none"> <li>3. El responsable de aseguramiento añade un nuevo artefacto a esa definición previamente creada.</li> <li>4. En el caso de que ya haya técnicas, roles, actividades o requisitos creados, podrá asociarlos al artefacto que acaba de crear, o bien modificar uno previamente creado.</li> </ol>
<b>Alternativas</b>	<ol style="list-style-type: none"> <li>1. Si se produce algún error a la hora de añadir o modificar cualquier artefacto o definición se avisará al responsable de aseguramiento mediante un mensaje por pantalla.</li> <li>2. El error aparecerá también en el <i>log</i> de la aplicación</li> </ol>
<b>Post-condiciones</b>	<ul style="list-style-type: none"> <li>• Se han creado nuevos artefactos o definiciones.</li> <li>• Esas modificaciones se ven reflejadas también en el avance del proyecto.</li> </ul>

Tabla 8: Caso de Uso "Gestionar Artefactos"

<b>Identificador</b>	CU-05
<b>Título</b>	Gestionar información de proceso.
<b>Actor</b>	Responsable de aseguramiento.
<b>Propósito</b>	Completar todo lo referido a las técnicas, roles y actividades del proyecto
<b>Descripción</b>	El responsable de aseguramiento deberá añadir las técnicas, roles y actividades necesarias para completar de forma satisfactoria el proyecto. Una vez añadidos, podrá modificarlos, asociarlos con otros elementos y borrarlos.
<b>Pre-condiciones</b>	<ul style="list-style-type: none"> <li>• La aplicación debe estar lanzada.</li> <li>• Se debe haber seleccionado un proyecto a través de la configuración de la aplicación.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. El responsable de aseguramiento selecciona un proyecto de la lista de proyectos creados.</li> <li>2. El responsable de aseguramiento selecciona uno de los <i>Reference Role</i>, <i>Reference Technique</i> o <i>Reference Activity</i> disponibles para añadir un nuevo elemento, según el caso que se corresponda ( técnicas, roles o actividades).</li> <li>3. En el caso de que ya haya artefactos creados, podrá</li> </ol>

	asociarlos al elemento que acaba de crear, o bien modificar uno previamente creado.
<b>Alternativas</b>	<ol style="list-style-type: none"> <li>1. Si se produce algún error a la hora de añadir o modificar cualquier artefacto o definición se avisará al responsable de aseguramiento mediante un mensaje por pantalla.</li> <li>2. El error aparecerá también en el <i>log</i> de la aplicación</li> </ol>
<b>Post-condiciones</b>	<ul style="list-style-type: none"> <li>• Se han creado nuevas técnicas, roles y/o actividades.</li> <li>• Esas modificaciones se ven reflejadas también en el avance del proyecto.</li> </ul>

Tabla 9: Caso de Uso "Gestionar Información de Proceso"

<b>Identificador</b>	CU-06
<b>Título</b>	Gestionar información de justificación.
<b>Actor</b>	Responsable de aseguramiento.
<b>Propósito</b>	Completar todo lo referido a los requisitos del proyecto.
<b>Descripción</b>	El responsable de aseguramiento deberá añadir los requisitos necesarios para completar de forma satisfactoria el proyecto. Una vez añadidos, podrá modificarlos, asociarlos con otros elementos y borrarlos.
<b>Pre-condiciones</b>	<ul style="list-style-type: none"> <li>• La aplicación debe estar lanzada.</li> <li>• Se debe haber seleccionado un proyecto a través de la configuración de la aplicación.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. El responsable de aseguramiento selecciona un proyecto de la lista de proyectos creados.</li> <li>2. El responsable de aseguramiento selecciona uno de los "Reference Requirements" disponibles para añadir un nuevo requisito.</li> <li>3. En el caso de que ya haya artefactos creados, podrá asociarlos al elemento que acaba de crear, o bien modificar uno previamente creado.</li> </ol>
<b>Alternativas</b>	<ol style="list-style-type: none"> <li>1. Si se produce algún error a la hora de añadir o modificar cualquier artefacto o definición se avisará al responsable de aseguramiento mediante un mensaje por pantalla.</li> </ol>

	2. El error aparecerá también en el <i>log</i> de la aplicación
<b>Post-condiciones</b>	<ul style="list-style-type: none"> <li>• Se han creado nuevos requisitos con los que se podrá trabajar.</li> <li>• Esas modificaciones se ven reflejadas también en el avance del proyecto.</li> </ul>

Tabla 10: Caso de Uso "Gestionar Información de Justificación"

## 3.2 Requisitos

A continuación se expondrán el propósito, alcance y visión general de la especificación de requisitos que se va a llevar a cabo.

### 3.2.1 Propósito y alcance

La especificación de requisitos mostrada a continuación analiza las funcionalidades que se incorporarán a la herramienta que se va a desarrollar.

El proyecto se divide fundamentalmente en dos bloques; de un lado, un repositorio de estándares de seguridad, que se visualizarán por pantalla junto a su cumplimentación. Del otro, un editor de los mismos.

### 3.2.2 Visión General

La especificación de requisitos que se va a exponer durante los siguientes apartados trata los siguientes puntos:

- Especificación de los requisitos de usuario.
- Especificación de los requisitos no funcionales.
- Especificación de los requisitos funcionales.
- Trazabilidad entre los requisitos y los casos de uso.

### 3.2.3 Modelo de exposición de los requisitos

Para la definición de los requisitos se va a utilizar el estándar IEEE-830 (IEEE Computer Society). La siguiente tabla muestra la estructura que utilizará en cada caso, rellenando cada apartado con la información correspondiente.

<b>Identificador</b>	
<b>Título</b>	
<b>Tipo</b>	
<b>Fuente del requisito</b>	
<b>Prioridad</b>	
<b>Descripción</b>	

Tabla 11: Formato de los requisitos

- Para el identificador del requisito se utilizará el siguiente formato:
  - Identificador: RXX YY.nn: R indica que se trata de un requisito, XX: será el tipo de requisito. Y nn: números consecutivos para la identificación del requisito concreto. YY podrá tomar los siguientes valores:
    - I: requisito de interfaz. En este caso, YY podrá valer:
      - U: de usuario.
      - H: de hardware.
      - SO: de software
    - F: requisitos funcionales. En este caso YY no tomará ningún valor.
    - NF: requisitos no funcionales. YY podrá tomar los siguientes valores:
      - R: de rendimiento.
      - F: de fiabilidad.
      - D: de disponibilidad.
      - P: de portabilidad.
- El campo nombre identifica, de manera resumida, el requisito.
  - El tipo podrá tomar los siguientes valores: Requisito o restricción.
- La fuente del requisito indica quien lo solicitó. Se diferenciará entre cliente y analista.
- La prioridad del requisito tomará uno de los siguientes valores:
  - Alta (esencial).
  - Media (deseado)
  - Baja (opcional).
- La descripción explicará de forma más detallada el requisito.

### 3.2.4 Requisitos funcionales

Especifican las funcionalidades que deberá cumplir el sistema que se va a desarrollar.

<b>Identificador</b>	RF 01
<b>Título</b>	Pantalla inicial de la aplicación
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	La aplicación deberá mostrar una pantalla inicial con las distintas opciones que se pueden llevar a cabo. Una vez que se abre la aplicación, se deberá mostrar un menú con las distintas opciones que hay. Estas son : Procesos, artefactos, proyecto, justificación y configuración. La información mostrada en ese menú de cada apartado será el título del mismo.

Tabla 12: Requisito Funcional "Pantalla inicial de la aplicación".

<b>Identificador</b>	RF 02
<b>Título</b>	Selección de proyectos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Se deberá seleccionar un proyecto, o bien añadir uno nuevo, en la pantalla "configuration" para poder ver los datos de la aplicación. En caso contrario la herramienta aparecerá vacía.

Tabla 13: Requisito Funcional "Selección de proyectos"

<b>Identificador</b>	RF 03
<b>Título</b>	Añadir un nuevo proyecto
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En el momento de crear un nuevo proyecto, el usuario deberá añadir un nombre identificativo y el modelo al que va a hacer



	referencia. En el caso de que no lo haga, la aplicación generará un error, se avisará al usuario y no se añadirá el proyecto. En caso contrario, se le notificará por pantalla de que el proyecto se ha creado correctamente.
--	---

Tabla 14: Requisito Funcional "Añadir un nuevo proyecto"

<b>Identificador</b>	RF 04
<b>Título</b>	Creación de nuevos <i>Artifact Definitions</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá crear todos los <i>Artifacts Definitions</i> que necesite. Para ello, tendrá que seleccionar el <i>Reference Artifact</i> del modelo al que va a hacer referencia y añadir sus datos básicos (nombre y descripción). En caso de que no se añada correctamente se le mostrará un error y se le re direccionará a la pantalla en la que se ha producido el fallo. Si se ha añadido de forma correcta, se le mostrará un mensaje afirmativo por pantalla.

Tabla 15: Requisito Funcional "Creación de nuevos *Artifact Definitions*"

<b>Identificador</b>	RF 05
<b>Título</b>	Añadir nuevos artefactos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Una vez creados los <i>Artifacts Definitions</i> , el usuario podrá añadir nuevos <i>Artifacts</i> a dicha definición. Para ello, deberá indicar el número de versión de la que se trata. En caso de que no se pueda añadir el artefacto, se le mostrará un error y se le re direccionará a la pantalla en la que se produjo para que vuelva a insertar los datos. Por el contrario, si se ha añadido correctamente, se les indicará mediante un mensaje por pantalla.

Tabla 16: Requisito Funcional "Añadir nuevos artefactos"

<b>Identificador</b>	RF 06
<b>Título</b>	Modificación de <i>Artifact Definitions</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En los casos en los que el usuario lo considere necesario, podrá modificar los datos introducidos al sistema sobre las definiciones. En caso de que se modifiquen correctamente se les mostrará por pantalla un mensaje indicándolo. En caso contrario, se les indicará también por pantalla y se les redirigirá al lugar donde se produjo el error.

Tabla 17: Requisito Funcional "Modificación de *Artifact Definitions*".

<b>Identificador</b>	RF 07
<b>Título</b>	Modificación de los artefactos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Una vez que está creado el artefacto, el usuario podrá modificar los datos asociados al mismo. En el caso de que se hayan producido las modificaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 18: Requisito Funcional "Modificación de los artefactos"

<b>Identificador</b>	RF 08
<b>Título</b>	Borrar <i>Artifact Definitions</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En el caso de que el usuario lo crea necesario, podrá borrar cualquier <i>Artifact Definition</i> que haya introducido. Con esta acción,

	se eliminarán también todos los artefactos que haya definido como sus hijos.
--	--

Tabla 19: Requisito Funcional "Borrar Artifact Definitions "

<b>Identificador</b>	RF 09
<b>Título</b>	Borrar artefactos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En el caso de que el usuario lo estime oportuno, podrá borrar cualquiera de los artefactos que haya introducido anteriormente, borrando con esta acción también todas las asociaciones al resto de elementos de la aplicación, tales como técnicas, requisitos , actividades, roles o sus eventos correspondientes.

Tabla 20: Requisito Funcional "Borrar artefactos"

<b>Identificador</b>	RF 10
<b>Título</b>	Ver detalles de un <i>Artifact Definition</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá visualizar los datos de los Artifact Definition que haya introducido en el sistema siempre que lo considere oportuno. Estos datos se actualizarán cada vez que se modifique un elemento.

Tabla 21: Requisito Funcional "ver detalles de un Artifact Definition"

<b>Identificador</b>	RF 11
<b>Título</b>	Ver detalles de un artefacto
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá visualizar los datos de los artefactos que haya

	introducido en el sistema siempre que lo considere oportuno. Estos datos se actualizarán cada vez que se modifiquen los datos referidos al elemento que se está viendo.
--	---

Tabla 22: Requisito Funcional "ver detalles de un artefacto"

<b>Identificador</b>	RF 12
<b>Título</b>	Añadir eventos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En el caso de que el usuario lo estime oportuno, podrá añadir tantos eventos referidos al artefacto con el que está trabajando como crea necesarios.

Tabla 23: Requisito Funcional "Añadir eventos."

<b>Identificador</b>	RF 13
<b>Título</b>	Tipos de eventos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los eventos que se podrán añadir a cada artefacto podrán ser de tipo "creación", "modificación", "evaluación" y "revocación".

Tabla 24: Requisito Funcional "Tipos de eventos".

<b>Identificador</b>	RF 14
<b>Título</b>	Añadir evaluación
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Para añadir una nueva evaluación a un artefacto, el usuario deberá seleccionar un participante de entre los existentes en la aplicación, que será el que la lleve a cabo.

Tabla 25: Requisito Funcional "Añadir evaluación."

<b>Identificador</b>	RF 15
<b>Título</b>	Creación de técnicas
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá crear todas las técnicas que necesite, referidas a los <i>Reference Technique</i> que se informan en el modelo, para completar su proyecto. En el caso de que se añadan de forma satisfactoria, se le mostrará un mensaje de confirmación por pantalla y, en caso contrario, se le mostrará un mensaje de error dándole la posibilidad de volver al punto donde se produjo el error.

Tabla 26: Requisito Funcional "Creación de técnicas"

<b>Identificador</b>	RF 16
<b>Título</b>	Modificación de técnicas
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En el caso en el que el usuario lo estime oportuno, podrá modificar los datos referidos a las técnicas que haya añadido anteriormente. En el caso de que el cambio se haga de forma satisfactoria, se le mostrará un mensaje por pantalla indicándoselo y, en caso contrario, se le mostrará un error, dándole la opción de volver al lugar de la aplicación en el que se produjo.

Tabla 27: Requisito Funcional "Modificación de técnicas"

<b>Identificador</b>	RF 17
<b>Título</b>	Borrado de técnicas
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta

<b>Descripción</b>	En caso de que el usuario necesite borrar una técnica de las que haya introducido anteriormente, podrá hacerlo, borrando con esta acción todas las asociaciones con el resto de componentes del proyecto.
--------------------	---

Tabla 28: Requisito Funcional "Borrado de técnicas"

<b>Identificador</b>	RF 18
<b>Título</b>	Ver detalles de una técnica
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá visualizar los datos las técnicas que haya introducido en el sistema siempre que lo considere oportuno. Estos datos se actualizarán cada vez que se modifique la técnica en cuestión.

Tabla 29: Requisito Funcional "ver detalles de una técnica"

<b>Identificador</b>	RF 19
<b>Título</b>	Añadir asociaciones entre artefactos y técnicas
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir todas las asociaciones que estime oportunas entre los artefactos y las técnicas que haya introducido al sistema según las restricciones de del modelo que esté usando como referencia. En el caso de que se hayan producido las asociaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 30: Requisito Funcional "añadir asociaciones entre artefactos y técnicas"

<b>Identificador</b>	RF 20
<b>Título</b>	Modificar asociaciones entre artefactos y técnicas
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá modificar todas las asociaciones que haya añadido entre los artefactos y las técnicas que haya introducido al sistema cuando lo necesite. En el caso de que se hayan producido las modificaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 31: Requisito Funcional "modificar asociaciones entre artefactos y técnicas"

<b>Identificador</b>	RF 21
<b>Título</b>	Creación de roles
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá crear todos los roles que necesite. En caso de que se añadan correctamente, se le informará al usuario mediante un mensaje en pantalla. En caso contrario, se le mostrará el error, dándole la posibilidad de volver al punto en el que se produjo.

Tabla 32: Requisito Funcional "Creación de roles"

<b>Identificador</b>	RF 22
<b>Título</b>	Modificación de roles
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá, modificar los datos de los roles añadidos al proyecto. En el caso de que el cambio se haga de forma satisfactoria, se le mostrará un mensaje por pantalla indicándoselo

	y, en caso contrario, se le mostrará un error, dándole la opción de volver al lugar de la aplicación en el que se produjo.
--	--

Tabla 33: Requisito Funcional "Modificación de roles"

<b>Identificador</b>	RF 23
<b>Título</b>	Borrado de roles
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En caso de que el usuario necesite borrar un rol de los que haya introducido anteriormente, podrá hacerlo, borrando con esta acción todas las asociaciones con el resto de componentes del proyecto.

Tabla 34: Requisito Funcional "Borrado de roles"

<b>Identificador</b>	RF 24
<b>Título</b>	Ver detalles de un rol
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá visualizar los datos los roles que haya introducido en el sistema siempre que lo considere oportuno. Estos datos se actualizarán cada vez que se modifique el participante en cuestión.

Tabla 35: Requisito Funcional "ver detalles de una técnica"

<b>Identificador</b>	RF 25
<b>Título</b>	Añadir asociaciones entre artefactos y roles
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir todas las asociaciones que estime oportunas entre los artefactos y los participantes que haya introducido al sistema según las restricciones de del modelo que



	esté usando como referencia. En el caso de que se hayan producido las asociaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.
--	--

Tabla 36: Requisito Funcional "añadir asociaciones entre artefactos y roles"

<b>Identificador</b>	RF 26
<b>Título</b>	Modificar asociaciones entre artefactos y roles
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá modificar todas las asociaciones que haya añadido entre los artefactos y los participantes que haya introducido al sistema cuando lo necesite. En el caso de que se hayan producido las modificaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 37: Requisito Funcional "modificar asociaciones entre artefactos y roles"

<b>Identificador</b>	RF 27
<b>Título</b>	Creación de actividades
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá crear todas las actividades que necesite, referidas a las <i>Reference Activity</i> del modelo con el que se esté trabajando. En caso de que se añadan correctamente, se le informará al usuario mediante un mensaje en pantalla. En caso contrario, se le mostrará otro mensaje con el error, dándole la posibilidad de volver al punto en el que se produjo.

Tabla 38: Requisito Funcional "Creación de actividades"..

<b>Identificador</b>	RF 28
<b>Título</b>	Modificación de actividades
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá, si es necesario, modificar los datos de las actividades añadidas al proyecto. En el caso de que el cambio se haga de forma satisfactoria, se le mostrará un mensaje por pantalla indicándoselo y, en caso contrario, se le mostrará un error, dándole la opción de volver al lugar de la aplicación en el que se produjo.

Tabla 39: Requisito Funcional "Modificación de actividades"

<b>Identificador</b>	RF 29
<b>Título</b>	Borrado de actividades
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En caso de que el usuario necesite borrar una actividad de las que haya introducido anteriormente, podrá hacerlo, borrando con esta acción todas las asociaciones con el resto de componentes del proyecto.

Tabla 40: Requisito Funcional "Borrado de actividades"

<b>Identificador</b>	RF 30
<b>Título</b>	Ver detalles de una actividad
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá visualizar los datos las actividades que haya introducido en el sistema siempre que lo considere oportuno. Estos datos se actualizarán cada vez que se modifique la actividad.

Tabla 41: Requisito Funcional "ver detalles de una actividad"

<b>Identificador</b>	RF 31
<b>Título</b>	Añadir asociaciones entre artefactos y actividades
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir todas las asociaciones que estime oportunas entre los artefactos y las actividades que haya introducido al sistema según las restricciones de del modelo que esté usando como referencia. En el caso de que se hayan producido las asociaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 42: Requisito Funcional "añadir asociaciones entre artefactos y actividades"

<b>Identificador</b>	RF 32
<b>Título</b>	Modificar asociaciones entre artefactos y actividades
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá modificar todas las asociaciones que haya añadido entre los artefactos y las actividades que haya introducido al sistema cuando lo necesite. En el caso de que se hayan producido las modificaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 43: Requisito Funcional "modificar asociaciones entre artefactos y actividades"

<b>Identificador</b>	RF 33
<b>Título</b>	Creación de requisitos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista

<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá crear todos los requisitos que necesite. En caso de que se añadan correctamente, se le informará al usuario mediante un mensaje en pantalla. En caso contrario, se le mostrará otro mensaje con el error, dándole la posibilidad de volver al punto en el que se produjo.

Tabla 44: Requisito Funcional "Creación de requisitos"

<b>Identificador</b>	RF 34
<b>Título</b>	Modificación de requisitos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá modificar los datos de los requisitos añadidos al proyecto. En el caso de que el cambio se haga de forma satisfactoria, se le mostrará un mensaje por pantalla indicándoselo y, en caso contrario, se le mostrará un error, dándole la opción de volver al lugar de la aplicación en el que se produjo.

Tabla 45: Requisito Funcional "Modificación de requisitos"

<b>Identificador</b>	RF 35
<b>Título</b>	Borrado de requisitos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	En caso de que el usuario necesite borrar un requisito de los que haya introducido, podrá hacerlo, borrando con esta acción todas las asociaciones con el resto de componentes del proyecto.

Tabla 46: Requisito Funcional "Borrado de requisitos"

<b>Identificador</b>	RF 36
<b>Título</b>	Ver detalles de un requisito
<b>Tipo</b>	Requisito

<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá visualizar los datos los requisitos que haya introducido en el sistema siempre que lo considere oportuno. Estos datos se actualizarán cada vez que se modifique el requisito en cuestión.

Tabla 47: Requisito Funcional "ver detalles de un requisito"

<b>Identificador</b>	RF 37
<b>Título</b>	Añadir asociaciones entre artefactos y requisitos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir todas las asociaciones que estime oportunas entre los artefactos y los requisitos que haya introducido al sistema según las restricciones de del modelo que esté usando como referencia. En el caso de que se hayan producido las asociaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo modo el error, y se le redirigirá a la pantalla en la que se produjo.

Tabla 48: Requisito Funcional "añadir asociaciones entre artefactos y requisitos"

<b>Identificador</b>	RF 38
<b>Título</b>	Modificar asociaciones entre artefactos y requisitos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá modificar todas las asociaciones que haya añadido entre los artefactos y los requisitos que haya introducido al sistema cuando lo necesite. En el caso de que se hayan producido las modificaciones correctamente, se le mostrará un mensaje afirmativo por pantalla. En caso contrario, se le indicará del mismo

	modo el error, y se le redirigirá a la pantalla en la que se produjo.
--	---

Tabla 49: Requisito Funcional "modificar asociaciones entre artefactos y requisitos"

<b>Identificador</b>	RF 39
<b>Título</b>	Ver progreso del proyecto
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá ver el avance del proyecto en todo momento, comprobando los porcentajes de elementos del modelo que han sido referenciados o completados durante el desarrollo del proyecto. Esta página se actualizará de manera automática a medida que se avanza en el desarrollo.

Tabla 50: Requisito Funcional "Ver progreso del proyecto"

<b>Identificador</b>	RF 40
<b>Título</b>	Creación de un nuevo <i>Reference Assurance Framework</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario tendrá que crear un <i>Reference Assurance Framework</i> cada vez que quiera introducir un modelo nuevo al proyecto. En caso de que se produzca algún error, este aparecerá por consola.

Tabla 51: Requisito Funcional "Creación de un nuevo *Reference Assurance Framework*"

<b>Identificador</b>	RF 41
<b>Título</b>	Modificación del <i>Reference Assurance Framework</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Si el usuario lo cree necesario, podrá modificar un <i>Reference Assurance Framework</i> previamente creado. Los cambios se verán reflejados tanto en la parte gráfica como no gráfica del mismo, en el caso de un componente modificable en el diagrama, y en la

	parte no gráfica en el caso de un componente no modificable mediante el diagrama.
--	---

Tabla 52: Requisito Funcional "Modificación del Reference Assurance Framework"

<b>Identificador</b>	RF 42
<b>Título</b>	Añadir nuevos <i>Reference Artifacts</i> al modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir tantos <i>Reference Artifacts</i> como considere necesario al modelo según el estándar de seguridad que esté siguiendo.

Tabla 53: Requisito Funcional "Añadir nuevos Reference Artifacts al modelo"

<b>Identificador</b>	RF 43
<b>Título</b>	Añadir nuevos <i>Reference Techniques</i> al modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir tantos <i>Reference Techniques</i> como considere necesario al modelo según el estándar de seguridad que esté siguiendo.

Tabla 54: Requisito Funcional "Añadir nuevos Reference Techniques al modelo"

<b>Identificador</b>	RF 44
<b>Título</b>	Añadir nuevos <i>Reference Roles</i> al modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir tantos <i>Reference Roles</i> como considere necesario al modelo según el estándar de seguridad que esté siguiendo.

Tabla 55: Requisito Funcional "Añadir nuevos Reference Roles al modelo"

<b>Identificador</b>	RF 45
<b>Título</b>	Añadir nuevos <i>Reference Activities</i> al modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir tantos <i>Reference Activities</i> como considere necesario al modelo según el estándar de seguridad que esté siguiendo.

Tabla 56: Requisito Funcional "Añadir nuevos *Reference Activities* al modelo"

<b>Identificador</b>	RF 46
<b>Título</b>	Añadir nuevos <i>Reference Requirements</i> al modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir tantos <i>Reference Requirements</i> como considere necesario al modelo según el estándar de seguridad que esté siguiendo.

Tabla 57: Requisito Funcional "Añadir nuevos *Reference Requirements* al modelo"

<b>Identificador</b>	RF 47
<b>Título</b>	Añadir nuevos <i>Reference Attribute</i> a los <i>Reference Artifacts</i> del modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir tantos <i>Reference Attribute</i> a los <i>Reference Artifacts</i> ya creados como considere necesario al modelo según el estándar de seguridad que esté siguiendo.

Tabla 58: Requisito Funcional "Añadir nuevos *Reference Attributes* al modelo"



<b>Identificador</b>	RF 48
<b>Título</b>	Añadir nuevas relaciones entre los <i>Reference Artifacts</i> del modelo
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá añadir relaciones entre los <i>Reference Artifacts</i> ya existentes como considere necesarias al modelo según el estándar de seguridad que esté siguiendo.

Tabla 59: Requisito Funcional "Añadir nuevas relaciones entre los *Reference Artifacts*"

<b>Identificador</b>	RF 49
<b>Título</b>	Borrado del <i>Reference Assurance Framework</i>
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El usuario podrá borrar un <i>Reference Assurance Framework</i> creado anteriormente.

Tabla 60: Requisito Funcional "Borrado del *Reference Assurance Framework*"

### 3.2.5 Requisitos de Interfaz

En este apartado se describen los requisitos de interfaz que se extraen del estudio de las necesidades. Se dividirán en tres grupos: Interfaces de los Usuarios, Interfaces Hardware e Interfaces Software.

#### 3.2.5.1 Interfaces de los Usuarios

Se especificará el aspecto visual que presentará el sistema en cuanto colores, forma y tamaño.

<b>Identificador</b>	RI U.01
<b>Título</b>	Un requisito por línea
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente

<b>Prioridad</b>	Alta
<b>Descripción</b>	Cada requisito quedará diferenciado del resto por líneas. No habrá dos requisitos juntos.

Tabla 61: Requisito de Interfaz de Usuario "Un requisito por línea"

<b>Identificador</b>	RI U.02
<b>Título</b>	Fondo de la interfaz
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente
<b>Prioridad</b>	Media
<b>Descripción</b>	El fondo de la interfaz será blanco.

Tabla 62: Requisito de Interfaz de Usuario "Fondo de la interfaz".

<b>Identificador</b>	RI U.03
<b>Título</b>	Botones de navegación
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente
<b>Prioridad</b>	Alta
<b>Descripción</b>	Se podrá navegar por las distintas pantallas a través de los botones de navegación.

Tabla 63: Requisito de Interfaz de Usuario "Botones de navegación"

<b>Identificador</b>	RI U.04
<b>Título</b>	Idioma
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente
<b>Prioridad</b>	Media
<b>Descripción</b>	Por defecto el idioma de la aplicación será el español.

Tabla 64: Requisito de Interfaz de Usuario "Idioma"

<b>Identificador</b>	RI U.05
<b>Título</b>	Botones de añadir y edición
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente

<b>Prioridad</b>	Media
<b>Descripción</b>	Deberá haber un botón a través del cual se permita la edición de algún elemento o su inserción en el proyecto.

Tabla 65: Requisito de Interfaz de Usuario "Botones de añadir y edición"

<b>Identificador</b>	RI U.06
<b>Título</b>	Estandarización de la interfaz.
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente
<b>Prioridad</b>	Alta
<b>Descripción</b>	La interfaz será la misma independientemente del usuario.

Tabla 66: Requisito de Interfaz de Usuario "Estandarización de la interfaz"

<b>Identificador</b>	RI U.07
<b>Título</b>	Logo.
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente
<b>Prioridad</b>	Media
<b>Descripción</b>	El logo con el nombre de la aplicación deberá estar visible en todo momento.

Tabla 67: Requisito de Interfaz de Usuario "Logo"

<b>Identificador</b>	RI U.08
<b>Título</b>	Botón guardar.
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Cliente
<b>Prioridad</b>	Media
<b>Descripción</b>	Deberá haber un botón que de la opción a guardar los cambios que se hayan producido.

Tabla 68: Requisito de Interfaz de Usuario "Botón guardar"

### 3.2.5.2 Interfaces Software

Se especifican las compatibilidades necesarias entre los diferentes programas y nuestra aplicación.

<b>Identificador</b>	RI SO.01
<b>Título</b>	Conexión a Base de Datos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Para que la aplicación funcione deberá estar conectada a una base de datos.

Tabla 69: Requisito de Interfaces Software "Conexión a base de datos"

<b>Identificador</b>	RI SO.02
<b>Título</b>	Servidor
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	La aplicación deberá estar alojada en un servidor.

Tabla 70: Requisito de Interfaces Software "Servidor"

### 3.2.5.3 Interfaces Hardware

Se describirán las características físicas de los dispositivos a utilizar.

<b>Identificador</b>	RI H.01
<b>Título</b>	Resolución de pantalla
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	La aplicación deberá adaptarse a la resolución de pantalla en la que se aloje.

Tabla 71: Requisito de Interfaces hardware "Resolución de pantalla"

<b>Identificador</b>	RI H.02
<b>Título</b>	Compatibilidad con navegadores
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	La aplicación deberá ser compatible con Internet Explorer, Mozilla Firefox y Google Chrome.

Tabla 72: Requisito de Interfaces Hardware "Compatibilidad con navegadores"

### 3.2.6 Requisitos no funcionales

Los requisitos no funcionales son aquellas restricciones para el producto una vez que ya se ha desarrollado. Se dividirán en: Rendimiento, fiabilidad, disponibilidad y portabilidad.

#### 3.2.6.1 Requisitos No Funcionales de Rendimiento

En este apartado se especifican aquellos requisitos relacionados con la carga que se espera pueda soportar el sistema.

<b>Identificador</b>	RNF R.01
<b>Título</b>	Usuarios simultáneos
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Sólo se podrá gestionar una petición al servidor al tiempo.

Tabla 73: Requisito No Funcional de Rendimiento "Usuarios simultáneos"

<b>Identificador</b>	RNF R.02
<b>Título</b>	Tiempo de reacción
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Media
<b>Descripción</b>	El tiempo de reacción a una petición al servidor no podrá ser mayor de 10 segundos.

Tabla 74: Requisito No Funcional de Rendimiento "Tiempo de reacción"

<b>Identificador</b>	RNF R.03
<b>Título</b>	Comunicación con la base de datos.
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	La comunicación de la aplicación con la base de datos deberá poder ser realizada en cualquier momento.

Tabla 75: Requisito No Funcional de Rendimiento "Comunicación con la base de datos"

<b>Identificador</b>	RNF R.04
<b>Título</b>	Tiempo de navegación.
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El tiempo de navegación entre pantallas, en condiciones normales, no deberá superar los dos segundos.

Tabla 76: Requisito No Funcional de Rendimiento "Tiempo de navegación"

### 3.2.6.2 Requisitos No Funcionales de Fiabilidad

Los siguientes requisitos están relacionados con la robustez del sistema y su comportamiento ante posibles fallos.

<b>Identificador</b>	RNF F.01
<b>Título</b>	Técnicas de prevención de errores SW
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Para la prevención de errores software se utilizarán lenguajes con abstracción de datos y modularidad, desarrollados en un entorno de pruebas de forma que se garantice su correcto funcionamiento antes de su puesta en producción.

Tabla 77: Requisito No Funcional de Fiabilidad "Técnicas de prevención de errores SW"

### 3.2.6.3 Requisitos No Funcionales de Disponibilidad

Se identificará la disponibilidad del sistema a los usuarios.

<b>Identificador</b>	RNF D.01
<b>Título</b>	Disponibilidad de la aplicación
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	La herramienta podrá estar disponible y ser utilizada en cualquier momento.

Tabla 78: Requisito No Funcional de Disponibilidad "Disponibilidad de la aplicación"

### 3.2.6.4 Requisitos No Funcionales de Portabilidad

Los siguientes requisitos tendrán que ver con las posibles migraciones de la aplicación a distintos equipos.

<b>Identificador</b>	RNF P.01
<b>Título</b>	Lenguaje utilizado en la aplicación.
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los lenguajes en los que se implementará la aplicación serán Java, HTML y JavaScript. De esta forma se garantizará la portabilidad del sistema.

Tabla 79: Requisito No Funcional de Portabilidad "Lenguaje utilizado en la aplicación"

<b>Identificador</b>	RNF P.02
<b>Título</b>	Código independiente del servidor.
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	El código en el que se implementará la aplicación será independiente del servidor que la aloje.

Tabla 80: Requisito No Funcional de Portabilidad "Código independiente del servidor"

<b>Identificador</b>	RNF P.03
<b>Título</b>	Lenguaje utilizado en la base de datos
<b>Tipo</b>	Restricción
<b>Fuente del requisito</b>	Analista
<b>Prioridad</b>	Alta
<b>Descripción</b>	Se trabajará con una base de datos MySQL, con el lenguaje Oracle.

Tabla 81: Requisito No Funcional de Portabilidad "Lenguaje utilizado en la base de datos"

### 3.3 Trazabilidad entre Casos de Uso y Requisitos Funcionales

En este apartado se mostrará la correspondencia entre cada caso de uso y cada requisito, de forma que sea fácilmente comprobable que se abarcan todas las opciones del proyecto.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06
RF-01		X				
RF-02		X				
RF-03		X				
RF-04				X		
RF-05				X		
RF-06				X		
RF-07				X		
RF-08				X		
RF-09				X		
RF-10				X		
RF-11				X		
RF-12				X		
RF-13				X		
RF-14				X		
RF-15					X	
RF-16					X	
RF-17					X	



RF-18					X	
RF-19				X		
RF-20				X		
RF-21					X	
RF-22					X	
RF-23					X	
RF-24					X	
RF-25				X		
RF-26				X		
RF-27					X	
RF-28					X	
RF-29					X	
RF-30					X	
RF-31				X		
RF-32				X		
RF-33						X
RF-34						X
RF-35						X
RF-36						X
RF-37				X		
RF-38				X		
RF-39			X			
RF-40	X					
RF-41	X					
RF-42	X					
RF-43	X					
RF-44	X					
RF-45	X					
RF-46	X					
RF-47	X					
RF-48	X					
RF-49	X					

Tabla 82: Trazabilidad entre casos de uso y requisitos

## 4. Diseño e Implementación

En este capítulo del documento se van a explicar las distintas alternativas barajadas para el desarrollo de la herramienta, así como la forma de implementarla.

### 4.1 Alternativas y decisiones de implementación

Primeramente se expondrán todas las opciones tenidas en cuenta a la hora del desarrollo de la aplicación, y seguidamente se explicarán las decisiones de la misma.

#### 4.1.1 Alternativas estudiadas

A continuación se explican las distintas alternativas tenidas en cuenta a la hora de desarrollar el proyecto. Este apartado se divide en tres epígrafes: Alternativas estudiadas para la creación del editor del modelo, alternativas para la creación de la base de datos y almacenamiento de los mismos, y alternativas de diseño gráfico de la aplicación.

##### 4.1.1.1 Alternativas para la creación del editor

Para crear el editor del modelo se tuvieron en cuenta las siguientes opciones:

- EMF Client: *Framework* para la creación de aplicaciones cliente basadas en modelos. Proporciona componentes para la interfaz que son reutilizables y extensibles para el desarrollo de aplicaciones basadas en un modelo dado.
- GMF: *The Eclipse Graphical Modeling Framework* genera los distintos componentes en tiempo de ejecución de los editores gráficos basados en EMF (*Eclipse Modeling Framework*).

Por motivos de eficiencia, después de ver el tipo de herramienta que se iba a desarrollar y de estudiar las dos opciones mostradas, se decidió que la mejor opción era crear el editor gráfico por un lado, a través de la segunda opción explicada, GMF, y el desarrollo de la interfaz de usuario por el otro.

En los puntos siguientes se explicará cómo se ha desarrollado esta parte del proyecto.

#### 4.1.1.2 Alternativas de almacenamiento

Para guardar o almacenar la información que se iba a manejar durante el uso de la aplicación se tuvieron en cuenta las siguientes alternativas:

- Generar un único esquema de base de datos para toda la aplicación, y poblarla según lo que el usuario decida introducir en su proyecto.
- Generar una base de datos y código específicos para cada proyecto, de forma que cada proyecto podrá tener una estructura distinta de base de datos.

Tras estudiar las dos alternativas expuestas, se ha decidido utilizar, por razones de eficiencia, la primera de las opciones. De esta forma, la aplicación será más sencilla y el almacenamiento de los datos menos complejo y con menos carga de tablas. Además, a la hora de gestionar un proyecto, no sólo habría que indicar los parámetros de la conexión a la base de datos tales como usuario y contraseña, sino que el usuario también debería introducir y cambiar, cada vez que se quiera cambiar de proyecto, la base de datos a la que hace referencia, lo que puede ser más complejo para el usuario final también, no solo desde el punto de vista del desarrollador.

En el siguiente apartado del documento se explicará cómo se ha llevado a cabo la creación de la base de datos que se va a utilizar.

#### 4.1.1.3 Alternativas de diseño

Por último, y en cuanto a las alternativas de diseño de la aplicación, se plantearon también dos opciones básicas:

- Creación de una aplicación similar a la de CRESCO, explicada en el segundo punto del documento, en la que se llevasen a cabo la transformación de los modelos y la gestión de los elementos desde un mismo sitio.
- Creación de una aplicación más visual y manejable que se adaptase a lo que el usuario introduzca en cada momento.

Una vez estudiadas ambas alternativas, se concluyó que la segunda opción era la mejor para este proyecto, debido a que el desarrollo de la primera es menos eficiente y menos flexible en cuanto a la información introducida, además de que la interfaz tiene limitaciones de uso.

En los siguientes puntos se explicará cómo se ha desarrollado finalmente la aplicación, y cómo sería un hipotético uso de la misma.

#### 4.1.2 Arquitectura software

Se ha decidido implementar la aplicación con una arquitectura software tipo MVC (*modelo-vista-controlador*). Esta arquitectura supone una separación entre las capas principales de un proyecto software:

- **Modelo:** Capa de la arquitectura donde se trabaja con los datos, conteniendo los mecanismos necesarios para acceder y modificar la información, es decir, todos los accesos a base de datos necesarios para el correcto funcionamiento de la aplicación.
- **Vista:** Capa de la arquitectura que facilita el contacto con el usuario. Contiene el código que permite la visualización de la herramienta.
- **Controlador:** Capa de la arquitectura que sirve de enlace entre la vista y el modelo. Contiene el código necesario para llevar a cabo las acciones y solicitudes de la aplicación.

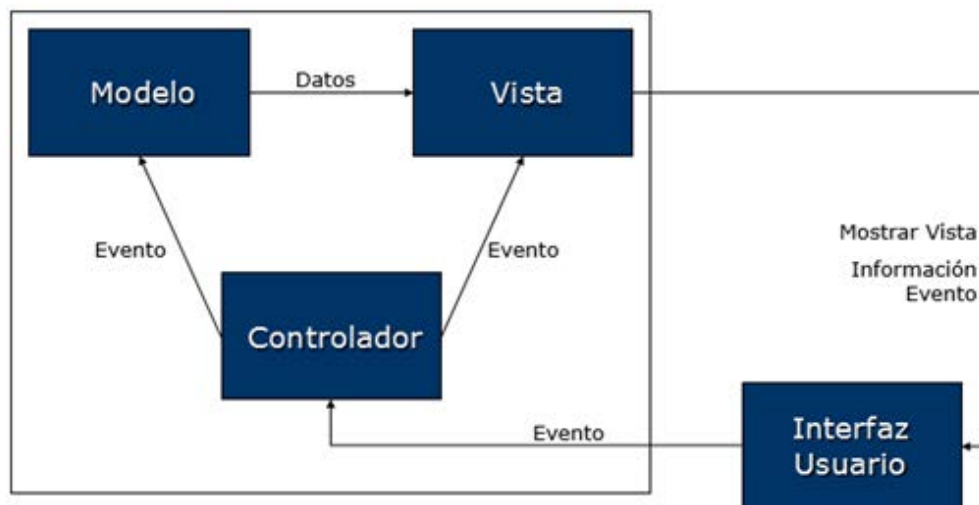


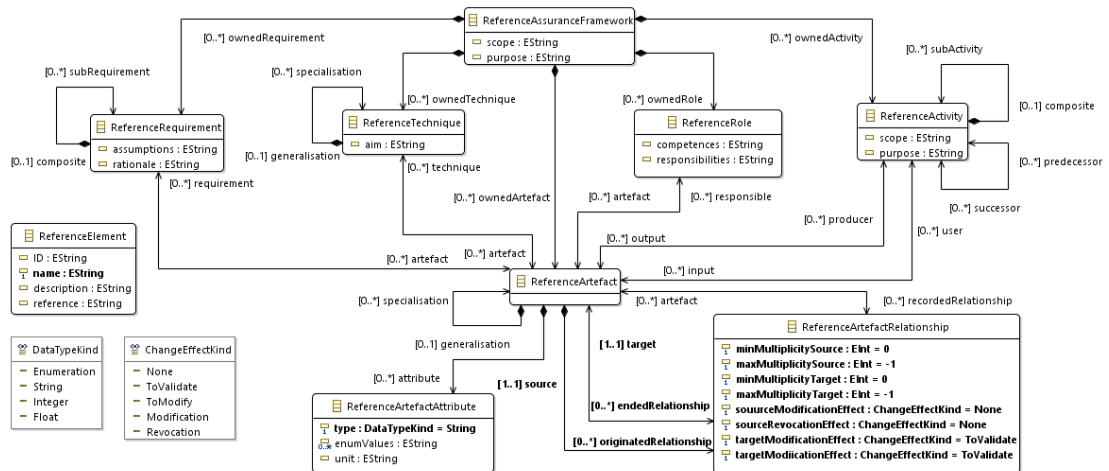
Ilustración 19: Esquema de la arquitectura Modelo-Vista-Controlador (V.Delgado, s.f.)

## 4.2 Implementación de la aplicación

La aplicación se ha desarrollado siguiendo, como se ha explicado en el apartado anterior, el esquema *modelo-vista-controlador*.

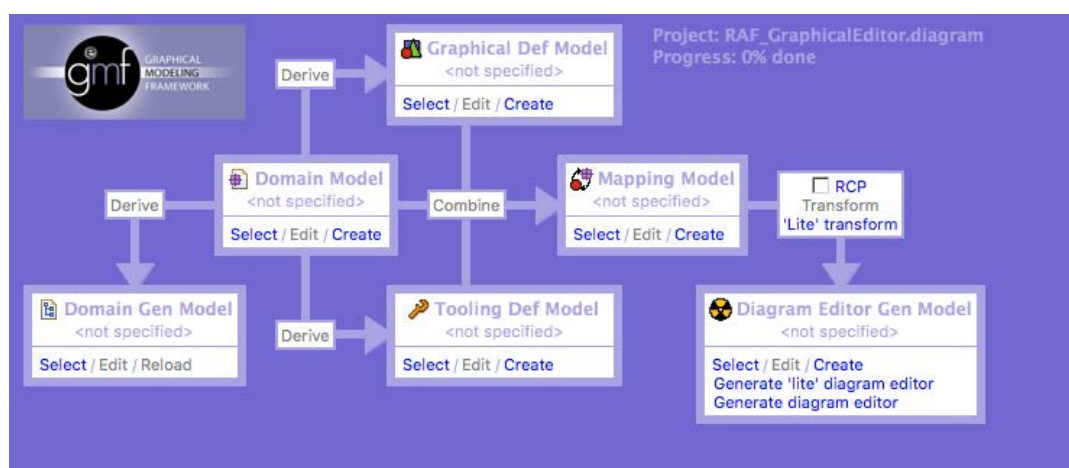
### 4.2.1 Editor gráfico

La creación del editor gráfico se ha hecho a través de la herramienta GMF de Eclipse, que se ha explicado anteriormente. Para realizarlo, se ha tenido en cuenta el siguiente meta-modelo:



*Ilustración 20: Meta-modelo utilizado para la generación del editor.*

La generación se realiza a raíz del fichero con extensión **.ecore**, que es el que contiene la definición del modelo. A partir de este, se crea un nuevo fichero con extensión **.genmodel**, a través del cual se comenzarán a crear el resto de elementos del editor. A partir de este fichero se generan las extensiones necesarias para crear el editor gráfico. La siguiente imagen muestra la pantalla a través de la cual se genera el editor:



*Ilustración 21: Pantalla de creación del GMF.*

- **Domain Model:** Primer paso para la creación del editor gráfico. En este punto se añade el ecore del modelo.
- **Domain Gen Model:** una vez introducido lo anterior, en este paso habrá que añadir el fichero `.genmodel` del modelo que se ha generado anteriormente a partir del `.ecore`.
- **Tooling Def Model:** en este punto se crea una paleta para crear el editor. En esta, simplemente se eligen las figuras que van a representar cada elemento de los que se van a poder añadir a través del editor gráfico.
- **Graphical Def Model:** en este punto se crea el fichero en el que se va a indicar como es la paleta con la que se va a crear el diagrama, se eligen los nodos, los componentes y las conexiones entre ellos. Se generará un nuevo fichero con extensión `.gmfgraph`.
- **Mapping Model:** combinando los últimos tres pasos explicados, se crea un mapa de representación. En este mapa, se indican todas las propiedades de cada elemento que se va a representar. Se indican como se dibujarán los nodos, las relaciones y los atributos de los nodos. Con todo esto, se creará un nuevo fichero con extensión `.gmfmap`.
- **Diagram Editor Gen Model:** una vez hecho esto, el último paso será transformar el mapa que se ha creado en un editor de diagramas a través de la opción “Transform” de la figura que se ha expuesto al principio.
- Una vez llevados a cabo los pasos anteriores, habrá que generar el código del nuevo diagrama a partir de la opción “Generate Diagram Code” del fichero `.gmfgen`.

Una vez creado todo lo anterior, al ejecutar el proyecto se obtendrá una pantalla como la siguiente. En ella, se podrán añadir los elementos de la parte gráfica y los elementos no gráficos al modelo:

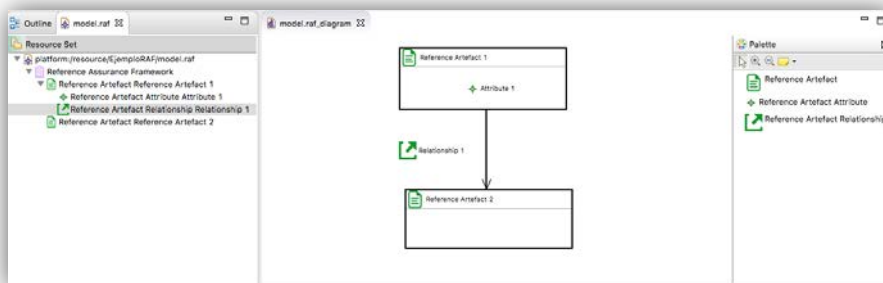


Ilustración 22: Pantalla principal de creación del modelo.

Como se puede ver en la imagen, la pantalla principal de creación se divide en tres secciones:

- En la barra de la derecha, se encuentran los elementos que se pueden añadir de forma gráfica, con los correspondientes iconos que se han elegido para cada elemento.
- En la barra de la izquierda, se ve el árbol que se crea cuando se añaden nuevos elementos al modelo. A medida que se van creando elementos gráficos, estos irán apareciendo en esta sección. El resto de elementos que no se crean de forma gráfica, se añadirán también a partir de esta lista.
- En la sección central, por último, es la parte en la que se crean las figuras, arrastrándolas desde el menú de la derecha. Al añadir una figura nueva, se podrán modificar las propiedades de la misma, y así añadirle los datos necesarios.

#### 4.2.2 Almacenamiento

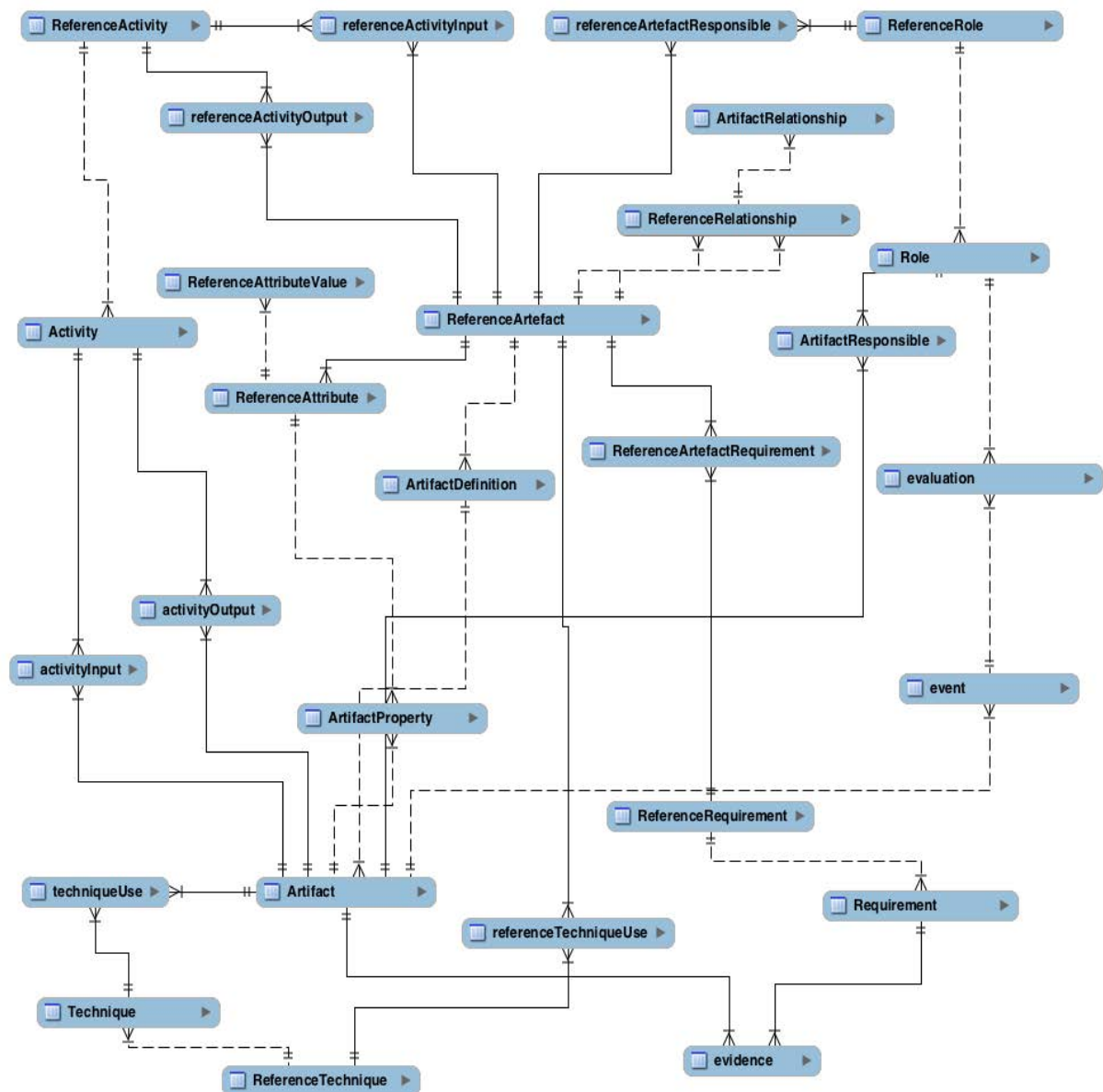
El almacenamiento de la información introducida a la aplicación se ha llevado a cabo a través de una base de datos tipo SQL relacional.

Para la creación de cada elemento de la misma, existían en principio dos alternativas: crear el diseño directamente desde el código de la herramienta o bien hacerlo de forma externa directamente a través del Framework de MySQL. Por no sobrecargar el código, se ha decidido hacerlo de la segunda forma citada. Desde el punto de vista de una futura entrega a clientes, la forma elegida también es la mejor alternativa, puesto que de esta forma, en el caso de necesitar realizar alguna modificación, se podrá hacer sin tener que abrir el código de la aplicación que, por lo general, queda encapsulado y compilado cuando se entrega.

El esquema de la base de datos se muestra a continuación. Debido a su extensión se han tomado las siguientes medidas a la hora de representarlo:

- Se han suprimido los atributos de las tablas.
- Se han suprimido las relaciones con la tabla "Project", y la misma, debido a que todas las tablas contienen una clave ajena a la misma, y la representación de la misma hace ilegible el diseño.

El esquema de la base de datos, con esto, es por tanto el siguiente:



*Ilustración 23: Esquema relacional de la base de datos.*

El script de creación de las tablas de la base de datos expuesta se encuentra en la carpeta “*Database*” de la aplicación, tal y como se expondrá en el quinto apartado del documento, referido al manual de uso de la herramienta.

A continuación se explican los significados y contenidos de cada una de las tablas expuestas en el diagrama:

- **ReferenceActivity:** Actividades de referencia del modelo que se ha cargado en la aplicación para llevar a cabo un proyecto. Contiene columnas para el id, nombre,



descripción, predecesor, composición, propósito, referencia, objetivo, sucesor y proyecto.

- **ReferenceArtefact:** Artefactos de referencia del modelo que se ha cargado como base para el desarrollo de un proyecto determinado. Contiene columnas para el id, nombre, descripción, atributo, generalización, relaciones, especialización y proyecto.
- **ReferenceRequirement:** Requisitos de referencia del modelo que se ha insertado en el programa para llevar a cabo un proyecto. Contiene columnas para su id, nombre, descripción, referencia, razón, asunciones, composición, sub requisitos y proyecto.
- **ReferenceTechnique:** Técnicas de referencia del modelo que se ha cargado en la herramienta para manejar un proyecto determinado. Contiene columnas para el id, nombre, descripción, referencia, objetivo, generalización, especialización y proyecto.
- **ReferenceRole:** Roles de referencia del modelo que se está utilizando en un proyecto determinado. Contiene columnas para el id, nombre, descripción, referencia, responsabilidades, competencias y proyecto.
- **ReferenceActivityInput:** Posibles relaciones que pueden darse entre las actividades y artefactos usuarios de las mismas introducidas por el usuario según el modelo que se está teniendo en cuenta. Contiene columnas para el id, el id del *Reference Artifact*, el id del *Reference Activity* y el proyecto.
- **ReferenceActivityOutput:** Posibles relaciones que pueden darse entre las actividades y artefactos productores de las mismas introducidas por el usuario según el modelo que se está teniendo en cuenta. Contiene columnas para el id, el id del *Reference Artifact*, el id del *Reference Activity* y el proyecto.
- **ReferenceArtefactRequirement:** Posibles relaciones que pueden darse entre los artefactos y requisitos que el usuario introduzca en su proyecto según el modelo

que se está teniendo en cuenta. Contiene columnas para el id, el id del *Reference Artifact*, el id del *Reference Requirement* y el proyecto.

- **ReferenceArtefactResponsible:** Posibles relaciones que pueden darse entre los artefactos y roles que el usuario introduzca en su proyecto según el modelo que se está teniendo en cuenta. Contiene columnas para el id, el id del *Reference Artifact*, el id del *Reference Role* y el proyecto.
- **ReferenceAttribute:** Posibles atributos que puede tomar el artefacto según el modelo con el que se está trabajando. Contiene columnas para el id, nombre, descripción, referencia, unidad, *Reference Artifact* al que se refiere, tipo y proyecto.
- **ReferenceAttributeValue:** Posibles atributos de tipo enumerado que puede tomar el artefacto según el modelo con el que se está trabajando en un momento dado. Contiene columnas para el id, id del *Reference Attribute* a que se refiere, el valor y el proyecto.
- **ReferenceRelationship:** Posibles relaciones entre artefactos que podrían darse en el proyecto según lo establecido en el modelo usado. Contiene columnas para el id, nombre, descripción, referencia, multiplicidades, artefacto con el que se relaciona, proyecto y artefacto del que nace la relación.
- **ReferenceTechniqueUse:** Posibles relaciones entre artefactos y técnicas que podrían darse en un proyecto según lo establecido en el modelo que se está usando para dicho proyecto. Contiene columnas para el id, el id del *Reference Artifact*, el id del *Reference Technique* y el proyecto.
- **Activity:** Distintas actividades que el usuario ha añadido a un determinado proyecto, y que están relacionadas con una determinada *Reference Activity* del modelo. Contiene columnas para el id, nombre, descripción, actividad de referencia, fecha de inicio y de fin de la actividad y proyecto.

- **ActivityInput:** Relaciones entre las actividades y los artefactos usuarios de las mismas en un proyecto determinado. Contiene columnas para el id, el id del artefacto, el id de la actividad y el proyecto.
- **ActivityOutput:** Relaciones entre las actividades y los artefactos productores de las mismas en un proyecto determinado. Contiene columnas para el id, el id del artefacto, el id de la actividad y el proyecto.
- **ArtifactDefinition:** Artefactos introducidos por el usuario para un determinado proyecto y que están relacionados con alguno de los *Reference Artifact* del modelo del estándar que se está tomando como referencia. Contiene columnas para el id, nombre, descripción, artefacto de referencia y proyecto.
- **Artifact:** Distintas versiones o variantes de un determinado *Artifact Definition* del proyecto. Contiene columnas para el id, nombre, descripción, definición a la que pertenece, versión, evaluación, artefacto anterior, localización, formato y proyecto.
- **ArtifactProperty:** Propiedades de un determinado artefacto del proyecto al que hacen referencia. Contiene columnas para el id, valor, atributo de referencia, artefacto al que pertenece y proyecto.
- **ArtifactRelationship:** Relaciones entre los distintos artefactos del proyecto. Contiene columnas para el id, el identificador del artefacto y del artefacto con el que se relaciona, nombre de la relación y proyecto.
- **ArtifactResponsible:** Relaciones entre los artefactos de un proyecto y los participantes o roles del mismo. Contiene columnas para el id, el id del artefacto, el id del rol y el proyecto.
- **Event:** Eventos asociados a cada artefacto de un determinado proyecto. Contiene columnas para el id, nombre, descripción, tipo de evento, hora a la que se ha llevado a cabo, artefacto al que se le añade el evento y proyecto.

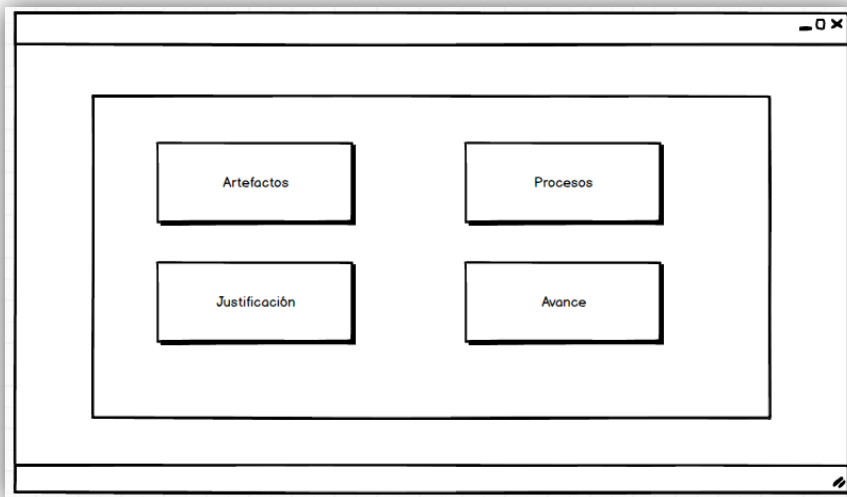
- **Evaluation:** Eventos de tipo evaluación de cada artefacto de los que existen en un proyecto concreto. Contiene columnas para el id, nombre, descripción, criterio, descripción del criterio, resultado de la misma, razón, participante que la lleva a cabo, evento al que se refiere y proyecto.
- **Evidence:** Relaciones existentes entre los distintos artefactos y requisitos de un determinado proyecto. Contiene columnas para el id, el id del artefacto, el id del requisito y el proyecto.
- **Project:** Proyectos en curso. Contiene columnas para el id, nombre y descripción del mismo.
- **Requirement:** Distintos requisitos que el usuario ha decidido introducir al proyecto con el que está trabajando, y que están relacionados con un determinado *Reference Requirement* del modelo. Contiene columnas para el id, nombre, descripción, requisito de referencia y proyecto.
- **Role:** Distintos roles que el usuario ha decidido introducir al proyecto con el que está trabajando, y que están relacionados con un determinado *Reference Role* del modelo. Contiene columnas para el id, nombre, descripción, email, rol de referencia y proyecto.
- **Technique:** Técnicas que el usuario ha introducido en un determinado proyecto y que están relacionadas con las *Reference Technique* del modelo que se está tomando como referencia. Contiene columnas para el id, nombre, descripción, técnica de referencia y proyecto.
- **TechniqueUse:** Relaciones existentes entre los distintos artefactos de un proyecto y las técnicas del mismo. Contiene columnas para el id, el id del artefacto, el id de la técnica y el proyecto.

### 4.2.3 Diseño

A continuación se muestran las interfaces iniciales de la aplicación. A medida que se fue añadiendo funcionalidad a la misma, su diseño fue variando hasta llegar al formato final, el cual se muestra en el apartado [5. Uso de la aplicación](#).

#### 4.2.3.1 Pantalla Principal

A través de la pantalla principal de la aplicación se puede llegar al resto de elementos de la misma. Esta pantalla constaba, en un principio, de cuatro enlaces, cada uno de los cuales redirigía al usuario a la parte de la aplicación a la que quería llegar. Para comenzar a utilizarla, el usuario debería entrar en alguno de los cuatro enlaces. Desde esta pantalla no se llevaba a cabo ninguna funcionalidad.



*Ilustración 24: interfaz inicial de la pantalla principal.*

El diseño actual de esta pantalla es muy similar, habiéndole añadido una nueva ventana que re dirige al usuario a la configuración de la aplicación, desde la cual podrá crear los proyectos nuevos con los que va a trabajar.



Ilustración 25: Interfaz final de la pantalla principal.

#### 4.2.3.2 Pantalla de Manipulación de Artefactos

Desde esta pantalla se llevaban a cabo todas las acciones relacionadas con los artefactos introducidos. En el menú de la izquierda se podían ver, en forma de árbol, los distintos elementos que se habían añadido a la herramienta. Seleccionando uno de ellos, aparecía en el cuadro principal de la pantalla lo referido al mismo, y se le daba al usuario la posibilidad de modificarlos. Se daba también, en un principio, la posibilidad de buscar directamente, mediante el cuadro de búsqueda, un artefacto concreto.

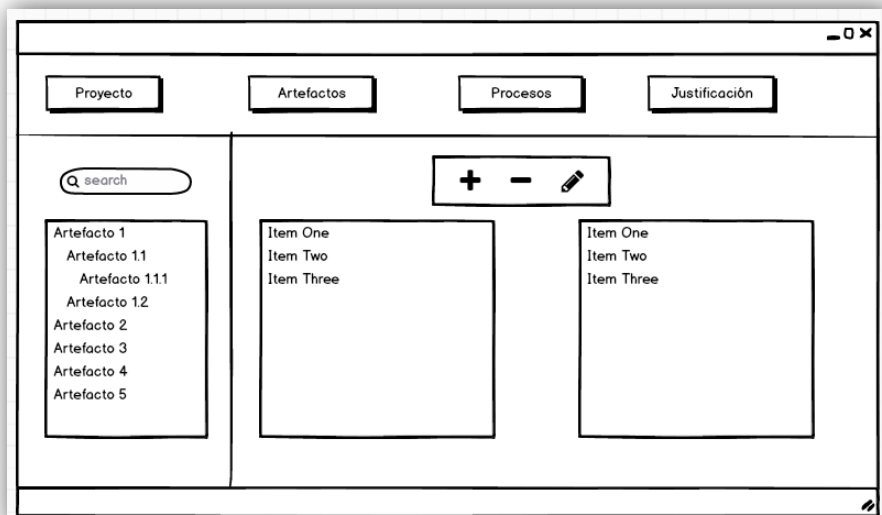


Ilustración 26: Interfaz inicial de la pantalla de manipulación de artefactos.

A la hora de llevar a cabo la implementación de la herramienta, se vio que era más útil otra forma de diseño, de forma que se mostrase en un primer momento cada *Reference Artifact* con sus correspondientes *Artifact Definition*. En caso de querer ver las distintas versiones de este *Artifact Definition*, se le da al usuario la opción de navegar mediante botones para llegar allí, desde donde podrá llevar a cabo las distintas tareas de modificación, asociación o creación.

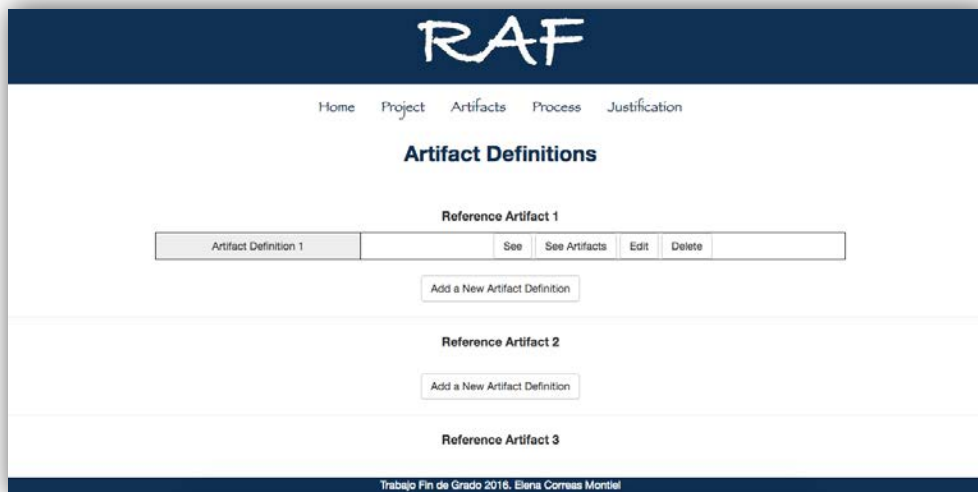


Ilustración 27: Interfaz final de la pantalla principal de manipulación de artefactos.

#### 4.2.3.3 Pantalla de Manipulación de Procesos

Desde esta pantalla se llevaban a cabo todas las acciones relacionadas con los procesos introducidos, es decir, técnicas, roles y actividades. En el menú de la izquierda se podían ver, en forma de árbol, los distintos elementos que se habían añadido a la herramienta. Seleccionando uno de ellos, aparecía en el cuadro principal de la pantalla lo referido al mismo, y se le daba al usuario la posibilidad de modificarlos. Se daba también, en un principio, la posibilidad de buscar directamente, mediante el cuadro de búsqueda, un elemento concreto.

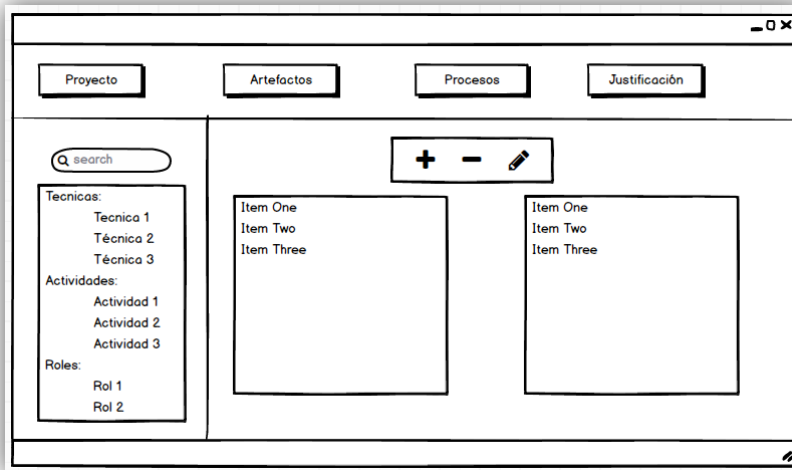


Ilustración 28: Diseño inicial de la pantalla de procesos

En el diseño final de esta pantalla, se ha dividido cada uno de los elementos que forman los procesos, de forma que existe un menú izquierdo en el que se puede seleccionar que tipo de elemento se quiere insertar o consultar. En la pantalla correspondiente a cada uno de los procesos, se pueden ver todos los Reference introducidos a través del modelo, y los distintos elementos para cada uno de ellos. Desde cada uno de estos elementos se podrá manipular su información.

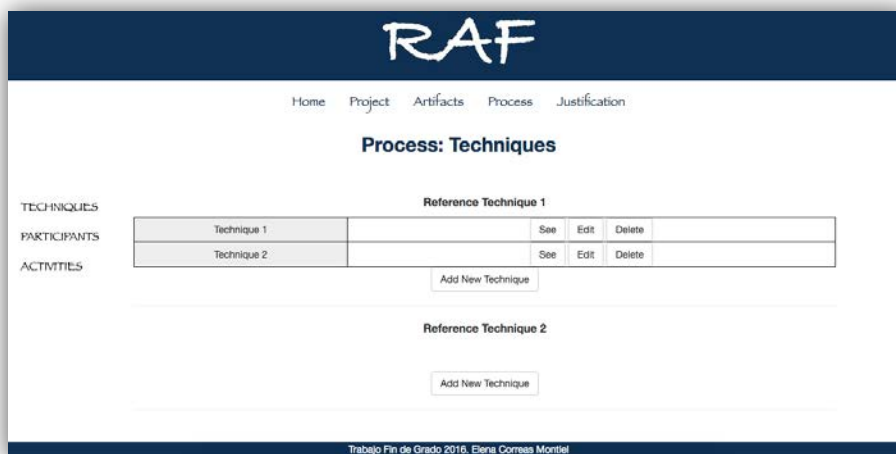


Ilustración 29: Interfaz final de la pantalla principal de manipulación de procesos.



#### 4.2.3.4 Pantalla de Manipulación de Requisitos

Desde esta pantalla se llevaban a cabo todas las acciones relacionadas con requisitos. En el menú de la izquierda se podían ver, en forma de árbol, los distintos elementos que se habían añadido a la herramienta. Seleccionando uno de ellos, aparecía en el cuadro principal de la pantalla lo referido al mismo, y se le daba al usuario la posibilidad de modificarlos. Se daba también, en un principio, la posibilidad de buscar directamente, mediante el cuadro de búsqueda, un requisito concreto.

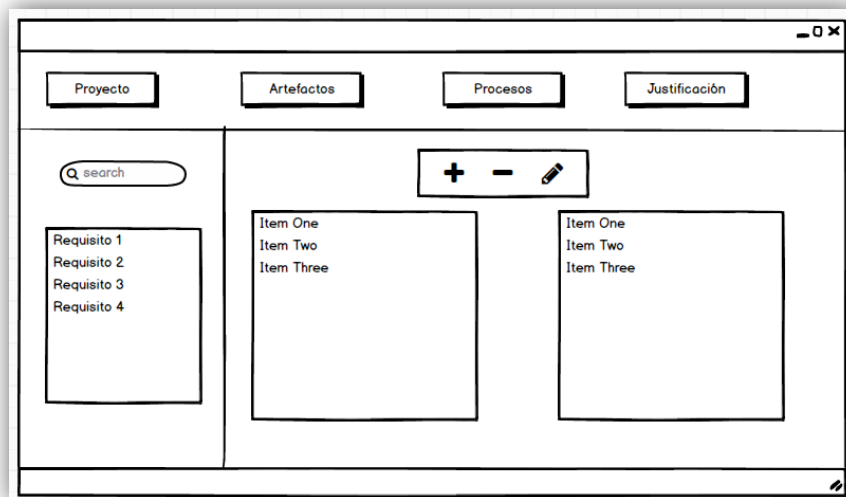


Ilustración 30: Diseño inicial de la pantalla de manipulación de requisitos.

La interfaz final implementada cumple las mismas características que las explicadas anteriormente. Desde la pantalla principal se pueden consultar todos los *Reference Requirements* que se han introducido con el modelo, y los requisitos asociados al mismo. A través de los botones de navegación se puede acceder a la manipulación de sus datos.

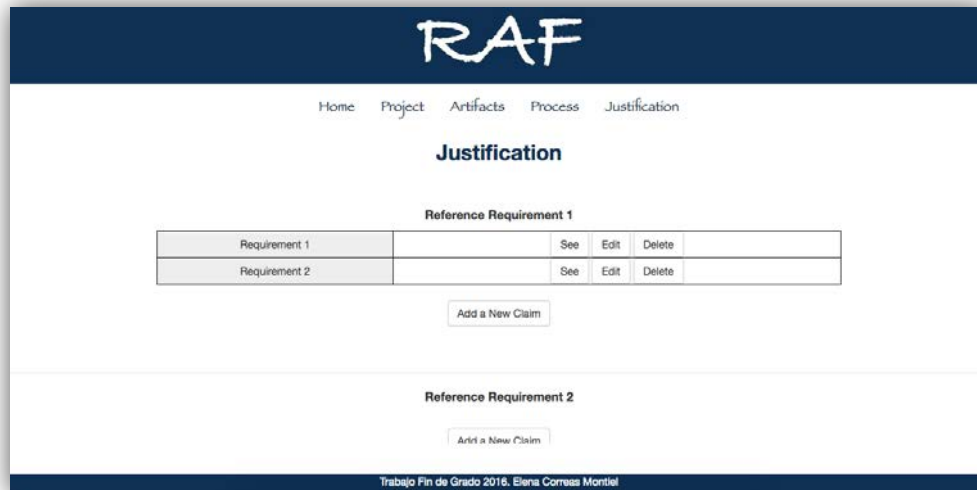


Ilustración 31: Interfaz final de la pantalla de manipulación de requisitos.

#### 4.2.3.5 Pantalla de Monitorización del proyecto

Por último, la pantalla de monitorización del proyecto trataba de mostrar el avance del proyecto en términos porcentuales. El diseño inicial era el siguiente:

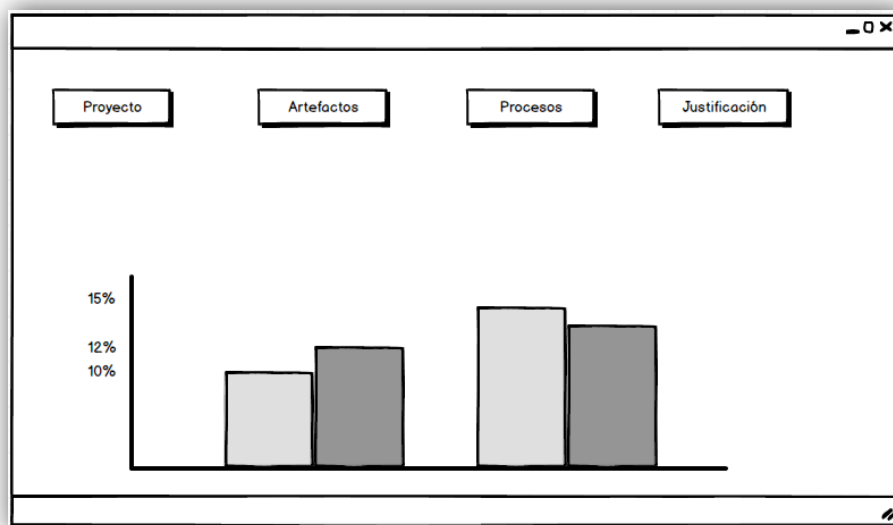


Ilustración 32: Interfaz inicial de la pantalla de monitorización del proyecto.

En la implementación final, se decidió tratar cada tipo de elemento por separado, mostrando una gráfica de avance para cada uno de ellos. El diseño, por lo tanto, quedó de la siguiente forma:

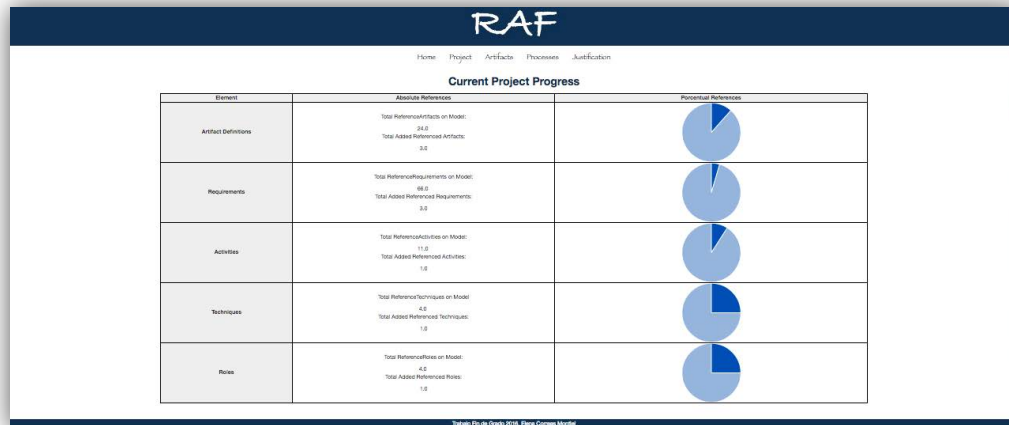


Ilustración 33: Interfaz final de la pantalla de monitorización del proyecto.

#### 4.2.4 Organización del proyecto

El proyecto está organizado de la siguiente forma:

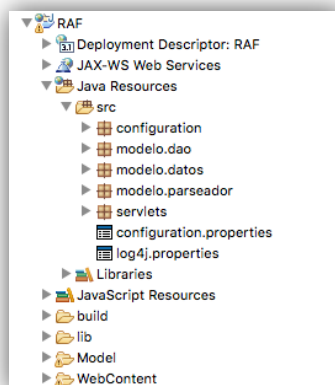


Ilustración 34: Organización del proyecto.

- **WebContent:** Carpeta donde se encuentra la parte de la “Vista” de la aplicación. Contiene todos los jsps, css y javascript que utiliza la herramienta.
- **Model:** Carpeta donde se almacenan automáticamente los modelos que se insertan en la aplicación.
- **Lib:** Contiene todas las librerías necesarias para el correcto funcionamiento de la herramienta.

- **Src/configuration:** Parte de la aplicación necesaria para configurar los parámetros de uso.
- **Src/modelo.dao:** Parte del código que contiene todos los accesos a base de datos. Contiene las funciones que van a permitir a la aplicación modificar, recuperar o añadir datos. Esta carpeta se corresponde con el “modelo” de la arquitectura.
- **Src/modelo.datos:** Parte del código que contiene las estructuras y atributos de cada uno de los elementos que se tienen en cuenta en la aplicación.
- **Src/modelo.parseador:** Contiene la parte del código que analiza y organiza el contenido del archivo que se sube a la aplicación como modelo de referencia.
- **Servlets:** Contiene todos los servlets necesarios para poder comunicar la interfaz de la aplicación con el acceso a los datos, es decir, la “vista” de la arquitectura con el “modelo”. Es, por tanto, el “controlador” de la arquitectura que se está usando.
- **Configuration.properties:** archivo de configuración en el que el usuario tendrá que introducir los datos privados necesarios para la conexión a la base de datos y la carga del modelo.
- **Log4j.properties:** archivo de configuración del *log* de la aplicación. El usuario deberá introducir la ruta en la que quiere que se genere dicho archivo.

El usuario se comunica con la aplicación a través de los archivos contenidos en la carpeta “WebContent”. Desde ella, según la acción que desee realizar el usuario, se llama a los distintos servlets contenidos en la carpeta “Servlets”. Desde estos, según lo necesario, se realiza la comunicación con los distintos archivos del resto de carpetas (“Src/modelo.dao”, “Src/modelo.datos”, “Src/modelo.parseador”).

#### 4.2.5 Diagrama de clases

En las siguientes imágenes se muestra el diagrama de clases por partes. Debido a su tamaño, se ha dividido en distintas partes, dejando juntas las clases entre las que existen relaciones. Así mismo, y para su mejor legibilidad, se ha decidido eliminar del diagrama los atributos y métodos que contienen cada una de las clases, siendo explicados detalladamente en su descripción.

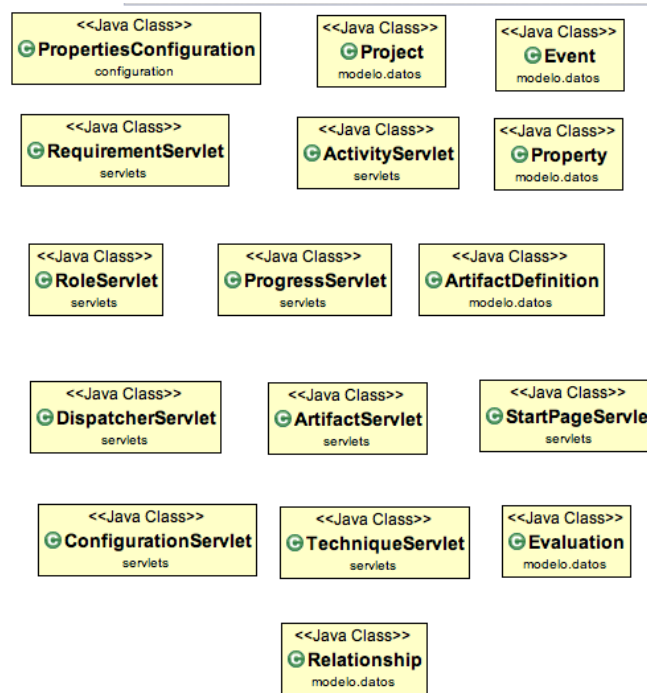


Ilustración 35: Diagrama de clases, parte I.

En esta primera imagen se ven aquellas clases que no tienen dependencias ni relaciones con el resto. Se puede diferenciar entre:

- **Project:** Clase que contiene los datos necesarios para almacenar el proyecto. Estos son, su id, nombre y descripción. En cuanto a sus métodos, cuenta con sus constructores y los *getters* y *setters* correspondientes a sus atributos.
- **ArtifactDefinition:** Contiene los datos correspondientes a las definiciones de los artefactos. Estos son los id, nombre, descripción, artefacto de referencia del modelo y proyecto al que pertenece. En cuanto a los métodos, en esta clase se encuentran los constructores de la misma y los *getters* y *setters* correspondientes a los atributos explicados.
- **Property:** Contiene los datos correspondientes a los atributos de un artefacto. Estos son, su valor, el identificador del atributo de referencia del modelo y el nombre del mismo atributo. Los métodos que contiene son los constructores de la clase y los *getters* y *setters* correspondientes a los atributos de la misma.

- **PropertiesConfiguration:** Clase que se encarga del análisis del fichero de configuración de la herramienta para así poder usar sus parámetros durante el resto del desarrollo y uso de la aplicación.
- **Event:** Clase que contiene los datos correspondientes a los eventos que se pueden llevar a cabo sobre un artefacto. Sus atributos son su id, nombre, descripción, tipo, hora a la que se ejecuta, identificador del artefacto al que se refiere y proyecto que lo contiene. Cuenta a su vez con los constructores de la clase y los *getters* y *setters* correspondientes.
- **Evaluation:** contiene los datos correspondientes al tipo de evento “evaluación”. Sus atributos son, su identificador, nombre, criterio de evaluación, descripción del criterio, resultado, razón, participante (persona que la lleva a cabo), evento al que corresponde y proyecto. Contiene también los constructores de la clase y los *getters* y *setters* correspondientes a los atributos.
- **Relationship:** contiene los datos que hay que tener en cuenta correspondientes a las relaciones que pueden darse entre los atributos. Estos son su identificador, nombre de la relación, artefactos que se relacionan y el identificador de la relación de referencia. En cuanto a los métodos, contiene los constructores de la clase y los *getters* y *setters* correspondientes a los atributos citados.
- **ConfigurationServlet:** Servlet que enlaza la vista correspondiente a la pantalla de configuración con las clases internas. Contiene los métodos *doGet()*, para recuperar los datos y mostrarlos por la interfaz, y el método *doPost()*, para recuperar los datos de entrada desde la vista y añadirlos al resto de elementos de la aplicación.
- **ActivityServlet:** Servlet que enlaza todas las vistas referidas a la creación y modificación de las actividades con la parte interna de la aplicación y sus accesos a la base de datos. Contiene los métodos *doGet()*, para recuperar los datos y mostrarlos por la interfaz, y el método *doPost()*, para recuperar los datos de entrada desde la vista y añadirlos al resto de elementos de la aplicación.
- **ArtifactServlet:** Servlet que enlaza todas las vistas referidas a la creación y modificación de los artefactos con la parte interna de la aplicación y sus accesos a la

base de datos. Contiene los métodos `doGet()`, para recuperar los datos y mostrarlos por la interfaz, y el método `doPost()`, para recuperar los datos de entrada desde la vista y añadirlos al resto de elementos de la aplicación.

- **TechniqueServlet:** Servlet que enlaza todas las vistas referidas a la creación y modificación de las técnicas con la parte interna de la aplicación y sus accesos a la base de datos. Contiene los métodos `doGet()`, para recuperar los datos y mostrarlos por la interfaz, y el método `doPost()`, para recuperar los datos de entrada desde la vista y añadirlos al resto de elementos de la aplicación.
- **RoleServlet:** Servlet que enlaza todas las vistas referidas a la creación y modificación de los roles con la parte interna de la aplicación y sus accesos a la base de datos. Contiene los métodos `doGet()`, para recuperar los datos y mostrarlos por la interfaz, y el método `doPost()`, para recuperar los datos de entrada desde la vista y añadirlos al resto de elementos de la aplicación.
- **DispatcherServlet:** Servlet que enlaza todas las vistas en las que se muestran los distintos elementos de la aplicación con la parte interna y sus respectivos accesos a la base de datos. Contiene el método `doGet()`, para recuperar los datos y mostrarlos por la interfaz.
- **RequirementServlet:** Servlet que enlaza todas las vistas referidas a la creación y modificación de los requisitos con la parte interna de la aplicación y sus accesos a la base de datos. Contiene los métodos `doGet()`, para recuperar los datos y mostrarlos por la interfaz, y el método `doPost()`, para recuperar los datos de entrada desde la vista y añadirlos al resto de elementos de la aplicación.
- **StartPageServlet:** Servlet que permite desplegar la aplicación. Contiene el método `doGet()`, que tiene como función desplegar el resto de elementos de la interfaz.
- **ProgressServlet:** Servlet que enlaza la vista que refleja el avance del proyecto con el resto de elementos internos de la aplicación. Contiene el método `doGet()` para recuperar esos datos.

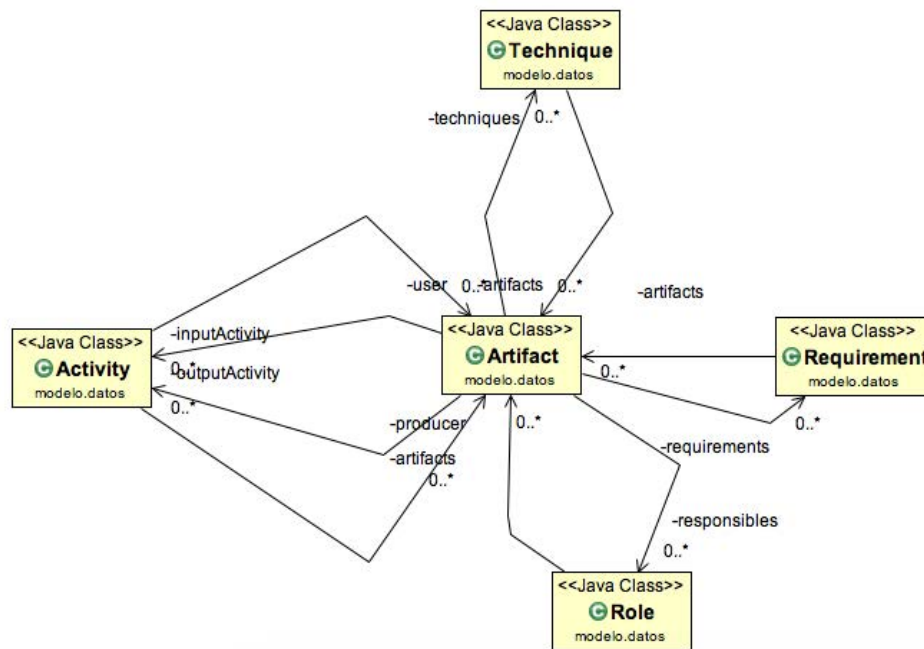


Ilustración 36: Diagrama de clases. Parte II.

Las clases que se muestran en esta figura son las siguientes:

- Artifact:** Clase que contiene los atributos de los artefactos. Estos son su id, nombre, descripción, definición, versión y proyecto al que pertenece. Además, se relaciona con las clases Roles, Técnicas, Requisitos y Actividades a través de las listas que contiene como atributos de éstos elementos. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- Activity:** Clase que contiene los atributos de las actividades. Estos son su id, nombre, descripción, actividad de referencia del modelo y proyecto al que pertenece. Además, se relaciona con la clase Artefactos a través de listas de tipo Artefacto, una para indicar los artefactos de los cuales es productor y otra para indicar los artefactos de los cuales es usuario. En cuanto a los métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- Requirement:** Clase que contiene los atributos de los requisitos. Estos son su id, nombre, descripción, requisito de referencia del modelo y proyecto al que pertenece. Además, se relaciona con la clase Artefactos a través de una Lista de artefactos. En cuanto a los métodos que contiene, estos son sus constructores de la clase, y los *getters* y *setters* correspondientes a cada atributo de la misma.



- **Role:** Clase que contiene los atributos de los roles. Estos son su id, nombre, descripción, rol de referencia del modelo y proyecto al que pertenece. Además, tiene también una lista de artefactos que indica cuáles de ellos están relacionados con cada uno de los roles que existen en el proyecto. En cuanto a los métodos que contiene, estos son sus constructores de la clase, y los *getters* y *setters* correspondientes a cada atributo de la misma.
- **Tecnhique:** Clase que contiene los atributos de las técnicas. Estos son su id, nombre, descripción, técnica de referencia del modelo que se está usando y proyecto al que pertenece. Además, tiene también una lista de artefactos que indica cuáles de ellos están relacionados con cada uno de las técnicas que existen en el proyecto. En cuanto a los métodos que contiene, estos son sus constructores de la clase, y los *getters* y *setters* correspondientes a cada atributo de la misma.
- En cuanto a las relaciones que se muestran, son las siguientes:
  - Un rol podrá pertenecer a varios ( o ningún) artefactos, y un artefacto tener varios (o ningún) roles. Por lo tanto las relaciones son de tipo 0..\*.
  - Una técnica podrá pertenecer a varios ( o ningún) artefactos, y un artefacto tener varias (o ninguna) técnica. Por lo tanto las relaciones son de tipo 0..\*.
  - Un requisito podrá pertenecer a varios ( o ningún) artefactos, y un artefacto tener varios (o ningún) requisito. Por lo tanto las relaciones son de tipo 0..\*.
  - Una actividad podrá ser el productor de varios ( o ningún) artefactos, y un artefacto podrá tener como productores varias (o ninguna) actividad. Por lo tanto las relaciones son de tipo 0..\*.
  - Una actividad podrá ser el usuario de varios ( o ningún) artefactos, y un artefacto podrá tener como usuarios varias (o ninguna) actividad. Por lo tanto las relaciones son de tipo 0..\*.

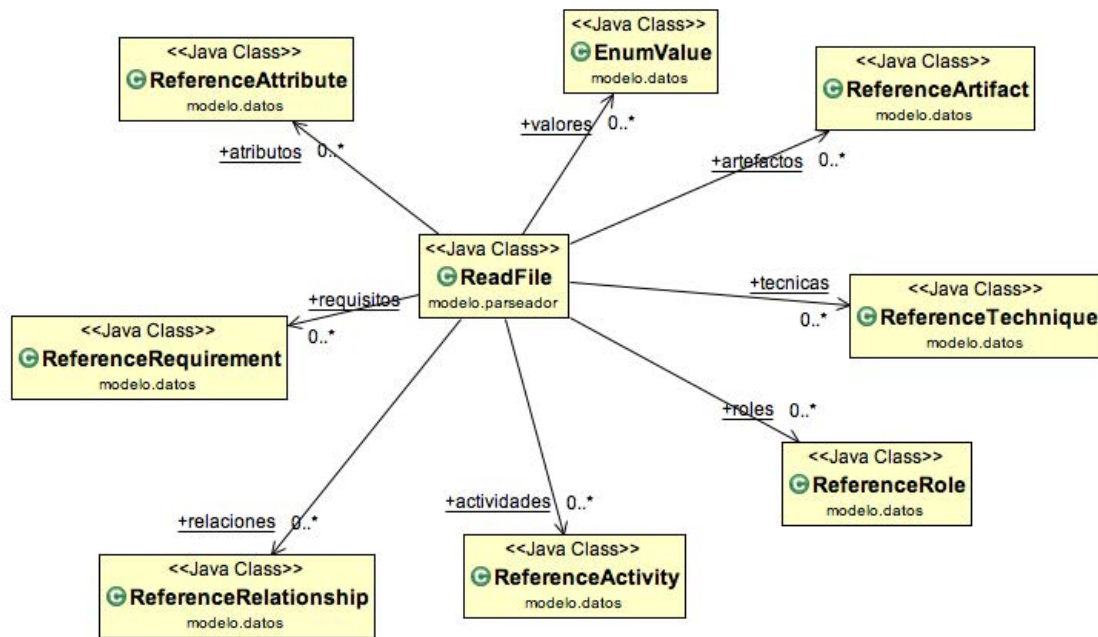


Ilustración 37: Diagrama de clases. Parte III.

En la imagen mostrada encima de estas líneas se ven las asociaciones entre las clases que implementan el estudio y análisis del archivo que se introduce en la herramienta como modelo, para su posterior inserción en la base de datos. Estas clases son las siguientes:

- **ReferenceRequirement:** Clase que contiene los atributos de los *Reference Requirement*. Estos son su id, nombre, descripción, referencia, razon, artefacto, asunciones, composición y sub-requisitos. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- **ReferenceActivity:** Clase que contiene los atributos de los *Reference Activity*. Estos son su id, nombre, descripción, inputs y outputs, predecesores, composiciones, propósito, referencia, objetivo y sucesor. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- **ReferenceAttribute:** Clase que contiene los atributos de los *Reference Attribute* de un *Reference Artifact*. Estos son su id, nombre, descripción, referencia, unidad, lista de valores y artefacto. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.

- **EnumValue:** Clase que contiene los atributos necesarios para almacenar los valores enumerados del atributo. Estos son el identificador del atributo y su valor. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- **ReferenceRole:** Clase que contiene los atributos de los *Reference Role*. Estos son su id, nombre, descripción, referencia, responsabilidades, competencias y artefacto. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- **ReferenceTechnique:** Clase que contiene los atributos de los *Reference Technique*. Estos son su id, nombre, descripción, referencia, propósito, generalización, especialización y artefacto. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- **ReferenceArtifact:** Clase que contiene los atributos de los *Reference Artefact*. Estos son su id, nombre, descripción, referencia, requisitos, atributos, generalizaciones, relaciones, responsables, especializaciones, técnicas y usuarios. Como métodos, contiene los constructores de la clase, y los *getters* y *setters* correspondientes a cada uno de sus atributos.
- **ReadFile:** Clase que contiene todos los métodos necesarios para hacer el análisis del fichero introducido por la aplicación. Los métodos que contiene son los siguientes:
  - **readContent (String, Integer):** Método que recibe como parámetros el archivo que se va a analizar y el identificador del proyecto al que va a hacer referencia. Según lo que vaya leyendo del fichero, llamará a unos u otros de los métodos mostrados a continuación.
  - **storeReferenceTechnique(String):** Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *Reference Technique* para su posterior inserción en base de datos.
  - **storeReferenceRole(String):** Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *Reference Role* para su posterior inserción en base de datos.

- ***storeReferenceRequirement(String)***: Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *Reference Requirement* para su posterior inserción en base de datos.
- ***storeReferenceAttribute(String)***: Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *Reference Attribute* para su posterior inserción en base de datos.
- ***storeReferenceArtefact(String)***: Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *Reference Artifact* para su posterior inserción en base de datos.
- ***storeReferenceActivity(String)***: Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *Reference Activity* para su posterior inserción en base de datos.
- ***storeEnumValues(String)***: Recibe por parámetro la línea del fichero que se va a analizar. Comprueba los datos que contiene la línea, y según lo que sea lo almacena a unos atributos u otros del *EnumValue* para su posterior inserción en base de datos.
- ***clasifyEnumValues(ArrayList<EnumValue>, ArrayList<ReferenceAttribute>)***: Analiza los *EnumValues* y los *Reference Attribute* almacenados para añadir los valores que se correspondan a cada atributo a la lista de valores enumerados de cada uno de ellos.

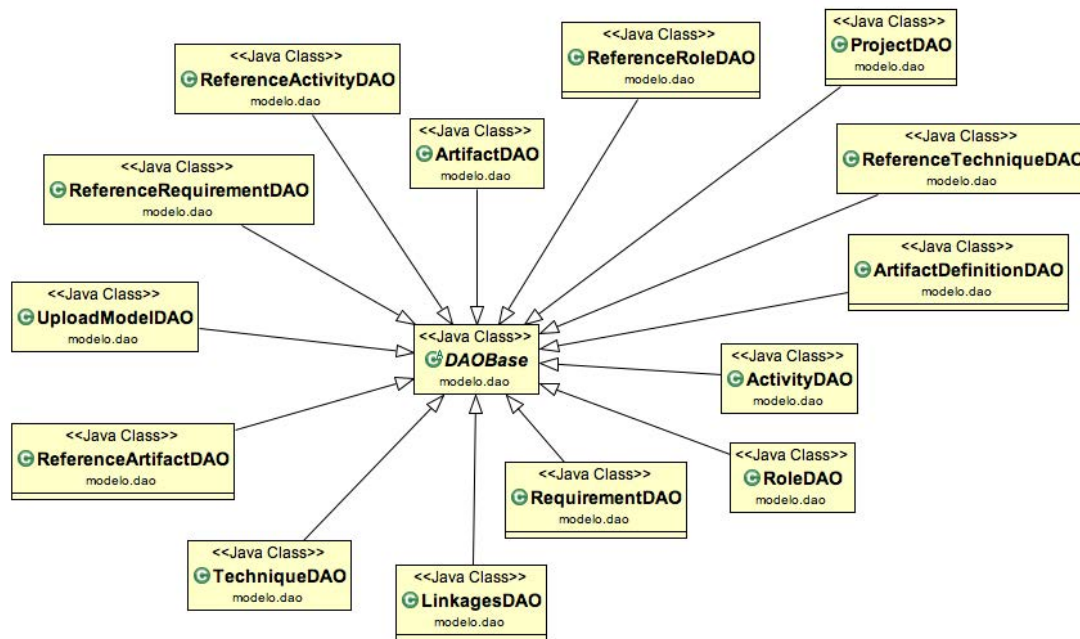


Ilustración 38: Diagrama de clases. Parte IV.

En la imagen anterior se ven las relaciones entre las clases que tienen como objetivo la manipulación de datos de la base de datos. A continuación se explican detenidamente cada una de ellas:

- **DAOBase:** Clase abstracta padre del resto de clases, y que contiene las definiciones de métodos que se utilizarán en las demás, según cual sea el elemento al que se refieran. Contiene los siguientes métodos:
  - **getAll(Integer):** Recibe por parámetro el proyecto en el que se encuentra el usuario. Su finalidad es la de obtener todos los registros de una tabla concreta.
  - **getConnection():** Método para establecer la conexión a la base de datos. Cada vez que cualquier método de cualquier clase realiza una acción sobre la base de datos, se llama a este método para recuperar dicha conexión.
  - **getCount(Integer):** Recibe por parámetro el id del proyecto con el que se está trabajando. Este método devuelve el número de registros que hay en una tabla determinada.
  - **getElement(Integer, Integer):** Recibe como parámetros el proyecto con el que se está trabajando y el id del elemento del que se quieren obtener los datos. Este método muestra todos los datos almacenados en una determinada tabla de la base de datos para un determinado elemento, que tendrá el id que recibe por parámetro.

- ***getTableName(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Se trata de un método abstracto que se implementa en cada una del resto de clases, y que devuelve el nombre de la tabla a la que acceder a realizar la consulta que se necesite.
- ***complete(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Se trata de un método abstracto que se encarga de recuperar los distintos datos que contenga cada tabla concreta.
- ***remove(Integer, Integer)***: Recibe como parámetros el id del proyecto con el que se está trabajando y el id del elemento que se quiere borrar. Este método será llamado cada vez que se quiera borrar un elemento concreto del proyecto.
- ***update (Object, Integer)***: Recibe como parámetro el objeto que se quiere modificar y el identificador del proyecto al que pertenece. Se trata de un método abstracto implementado en cada una de las distintas clases y que permite actualizar un registro determinado de la base de datos.
- **UploadModelDAO**: Clase que se encarga de todos los accesos a base de datos para hacer la carga del modelo de referencia que se va a usar para realizar un proyecto. Consta de los siguientes métodos:
  - ***uploadActivity(ReferenceActivity, Integer)***: Recibe por parámetro el “ReferenceActivity” obtenido de la lectura del fichero y el identificador del proyecto al que hay que añadirlo. Inserta en la base de datos todos los “ReferenceActivity” que se han examinado.
  - ***uploadActivityInput(Integer, Integer, Integer)***: Recibe por parámetro el identificador de una actividad, del artefacto con el que tiene relación y el proyecto con el que se está trabajando. Insertar en la base de datos las relaciones entre las actividades de entrada (productores) de los artefactos y dichos artefactos.
  - ***uploadActivityOutput(Integer, Integer, Integer)***: Recibe por parámetro el identificador de una actividad, del artefacto con el que tiene relación y el proyecto con el que se está trabajando. Insertar en la base de datos las relaciones entre las actividades de salida (usuarios) de los artefactos y dichos artefactos.
  - ***uploadArtefact(ReferenceArtefact, Integer)***: Recibe por parámetro el *Reference Artefact* obtenido de la lectura del fichero y el identificador del

proyecto al que hay que añadirlo. Inserta en la base de datos todos los *Reference Artifact* que se han examinado.

- ***uploadArtefactRequirement (Integer, Integer, Integer)***: Recibe por parámetro el identificador de una requisito, del artefacto con el que tiene relación y el proyecto con el que se está trabajando. Insertar en la base de datos las relaciones entre dichos requisitos y artefactos.
- ***uploadArtefactResponsible (Integer, Integer, Integer)***: Recibe por parámetro el identificador de una rol, del artefacto con el que tiene relación y el proyecto con el que se está trabajando. Insertar en la base de datos las relaciones entre dichos roles y artefactos.
- ***uploadAttribute(ReferenceAttribute, Integer)***: Recibe por parámetro el *Reference Attribute* obtenido de la lectura del fichero y el identificador del proyecto al que hay que añadirlo. Inserta en la base de datos todos los *Reference Attribute* que se han examinado.
- ***uploadAttributeValues(Integer, String, Integer)***: Recibe por parámetro el identificador de una atributo, el valor del mismo y el proyecto con el que se está trabajando. Insertar en la base de datos las relaciones entre dichos atributos y sus posibles valores.
- ***uploadRequirement(ReferenceRequirement, Integer)***: Recibe por parámetro el *Reference Requirement* obtenido de la lectura del fichero y el identificador del proyecto al que hay que añadirlo. Inserta en la base de datos todos los *Reference Requirement* que se han examinado.
- ***uploadRole(ReferenceRole, Integer)***: Recibe por parámetro el *Reference Role* obtenido de la lectura del fichero y el identificador del proyecto al que hay que añadirlo. Inserta en la base de datos todos los *Reference Role* que se han examinado.
- ***uploadTechnique(ReferenceTechnique, Integer)***: Recibe por parámetro el *Reference Technique* obtenido de la lectura del fichero y el identificador del proyecto al que hay que añadirlo. Inserta en la base de datos todos los *Reference Technique* que se han examinado.
- ***uploadTechniqueUse(Integer, Integer, Integer)***: Recibe por parámetro el identificador de una técnica, del artefacto con el que tiene relación y el proyecto con el que se está trabajando. Insertar en la base de datos las relaciones entre dichas técnicas y artefactos.

- ***uploadRelationship(ReferenceRelationship, Integer)***: Recibe como parámetro la relación a almacenar y el identificador del proyecto con el que se está trabajando. Inserta en la base de datos dicha relación con sus atributos correspondientes.
- **ReferenceActivityDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada *Reference Activity* almacenado en la base de datos. Consta de los siguientes métodos:
  - ***getNombreTabla(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos *Reference Activity*.
  - ***rellenar(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto *Reference Activity* con los datos obtenidos de la consulta a base de datos.
- **ReferenceRoleDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada *Reference Role* almacenado en la base de datos. Consta de los siguientes métodos:
  - ***getNombreTabla(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos *Reference Role*
  - ***rellenar(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto *Reference Role* con los datos obtenidos de la consulta a base de datos.
- **ReferenceArtefactDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada *Reference Artifact* almacenado en la base de datos. Consta de los siguientes métodos:
  - ***getNombreTabla(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos *Reference Artifact*.
  - ***rellenar(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto *Reference Artifact* con los datos obtenidos de la consulta a base de datos.



- **ReferenceRequirementDAO:** Clase que contiene todos los métodos encargados de la manipulación de los datos de cada *Reference Requirement* almacenado en la base de datos. Consta de los siguientes métodos:
  - **getNombreTabla(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos *Reference Requirement*.
  - **rellenar(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto *Reference Requirement* con los datos obtenidos de la consulta a base de datos.
- **ReferenceTechniqueDAO:** Clase que contiene todos los métodos encargados de la manipulación de los datos de cada *Reference Technique* almacenado en la base de datos. Consta de los siguientes métodos:
  - **getNombreTabla(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos *Reference Technique*.
  - **rellenar(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto *Reference Technique* con los datos obtenidos de la consulta a base de datos.
- **ArtifactDAO:** Clase que contiene todos los métodos encargados de la manipulación de los datos de cada “Artefacto” almacenado en la base de datos. Consta de los siguientes métodos:
  - **getActivityInput(Integer, Integer):** Recibe como parámetros el identificador del artefacto y el identificador del proyecto en curso. Devuelve una lista de actividades que podrían ser asociadas como productoras con ese artefacto según las restricciones del modelo que se está siguiendo.
  - **getActivityOutput(Integer, Integer):** Recibe como parámetros el identificador del artefacto y el identificador del proyecto en curso. Devuelve una lista de actividades que podrían ser asociadas como usuarios con ese artefacto según las restricciones del modelo que se está siguiendo.
  - **getAllArtefacts(Integer, Integer):** Recibe como parámetro el identificador del *Artifact Definition* al que hace referencia y el identificador del proyecto con el que se está trabajando. Devuelve una lista de artefactos referidos al *Artifact Definition* que recibe como parámetro.

- **getTableName(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos artefactos.
  - **insertArtifact(Artefactos, Integer):** Recibe por parámetro el artefacto que ha introducido el usuario en la aplicación y el proyecto con el que se está trabajando. Inserta en la base de datos un nuevo registro con ese artefacto recientemente añadido.
  - **complete(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto “Artefactos” con los datos obtenidos de la consulta a base de datos.
  - **update(Object, Integer):** Recibe como parámetro el artefacto que se quiere modificar y el identificador del proyecto al que pertenece. Modifica los datos referidos al artefacto que se le indica.
  - **getPreviousArtifact(Integer, Integer):** Recibe como parámetro el identificador del artefacto actual y del proyecto con el que se está trabajando, y devuelve el artefacto anterior al nuevo.
  - **getArtifactDefinition(Integer, Integer):** Recibe como parámetro el identificador del artefacto actual y del proyecto con el que se está trabajando, y devuelve el *artifact definition* al que corresponde el artefacto actual.
- **ActivityDAO:** Clase que contiene todos los métodos encargados de la manipulación de los datos de cada “Actividad” almacenada en la base de datos. Consta de los siguientes métodos:
    - **countReferences(Integer):** Recibe por parámetro el proyecto con el que está trabajando el usuario. Devuelve el número total de actividades que han sido asociadas a los *Reference Activity* que existen en el proyecto.
    - **getArtifactProducer(Integer, Integer):** Recibe por parámetro el identificador de una actividad concreta y el identificador del proyecto en curso. Devuelve una lista de artefactos que, según el modelo añadido al proyecto, se han asociado a una actividad dada como productores.
    - **getNOArtifactProducer(Integer, Integer):** Recibe por parámetro el identificador de una actividad concreta y el identificador del proyecto en curso. Devuelve una lista de artefactos que, según el modelo añadido al proyecto, aun no se han asociado a una actividad dada como productores.

- ***getReferenceActivity(Integer, Integer)***: Recibe por parámetro el identificador de una actividad y el proyecto al que pertenece. Devuelve el *Reference Activity* al que hace referencia.
- ***getArtifactUser(Integer, Integer)***: Recibe por parámetro el identificador de una actividad concreta y el identificador del proyecto en curso. Devuelve una lista de artefactos que, según el modelo añadido al proyecto, se han asociado a una actividad dada como usuarios.
- ***getNOArtifactProducer(Integer, Integer)***: Recibe por parámetro el identificador de una actividad concreta y el identificador del proyecto en curso. Devuelve una lista de artefactos que, según el modelo añadido al proyecto, aun no se han asociado a una actividad dada como usuarios.
- ***getTableName(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan las distintas *actividades*.
- ***insertActivityInput(Artefactos, Actividades, Integer)***: Recibe como parámetros el artefacto, actividad, y proyectos con los que se está trabajando. Inserta en la base de datos la nueva asociación entre dichos artefactos y actividades de entrada.
- ***insertActivityOutput(Artefactos, Actividades, Integer)***: Recibe como parámetros el artefacto, actividad, y proyectos con los que se está trabajando. Inserta en la base de datos la nueva asociación entre dichos artefactos y actividades de salida.
- ***insertActivity(Actividades, Integer)***: Recibe por parámetro la actividad que ha introducido el usuario en la aplicación y el proyecto con el que se está trabajando. Inserta en la base de datos un nuevo registro con esa actividad recientemente añadida.
- ***complete(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto “Actividades” con los datos obtenidos de la consulta a base de datos.
- ***update(Object, Integer)***: Recibe como parámetro la actividad que se quiere modificar y el identificador del proyecto al que pertenece. Modifica los datos referidos a la actividad que se le indica.
- ***removeActivityInput(Activity, Integer)***: recibe por parámetro la actividad de la que se quieren borrar las relaciones y el proyecto al que pertenece. Borra una asociación entre actividades y artefactos.

- ***removeActivityInput(Activity, Integer)***: recibe por parámetro la actividad de la que se quieren borrar las relaciones y el proyecto al que pertenece. Borra una asociación entre actividades y artefactos.
- **ArtifactDefinitionDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada *Artefact Definition* almacenado en la base de datos. Consta de los siguientes métodos:
  - ***countReferences(Integer)***: Recibe por parámetro el proyecto con el que está trabajando el usuario. Devuelve el número total de *Artifact Definition* que han sido asociados a los *Reference Artifact* que existen en el proyecto.
  - ***insertArtifactDefinition(ArtifactDefinition, Integer)***: Recibe por parámetro el *ArtefactDefinition* que ha introducido el usuario en la aplicación y el proyecto con el que se está trabajando. Inserta en la base de datos un nuevo registro con ese *Artifact Definition* recientemente añadido.
  - ***getTableName(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos *Artifact Definition*
  - ***complete(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto *Artifact Definition* con los datos obtenidos de la consulta a base de datos.
  - ***update(Object, Integer)***: Recibe como parámetro el *Artifact Definition* que se quiere modificar y el identificador del proyecto al que pertenece. Modifica los datos referidos al que se le indica.
  - ***getReferenceArtifact(Integer, Integer)***: Recibe por parámetro el identificador de un artefacto y el proyecto al que pertenece. Devuelve el nombre del *Reference Artifact* al que hace referencia.
- **RequirementDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada requisito almacenado en la base de datos. Consta de los siguientes métodos:
  - ***countReferences(Integer)***: Recibe por parámetro el proyecto con el que está trabajando el usuario. Devuelve el número total de requisitos que han sido asociados a los *Reference Requirement* que existen en el proyecto.
  - ***insertRequirement(Requisitos, Integer)***: Recibe por parámetro el requisito que ha introducido el usuario en la aplicación y el proyecto con el que se está

trabajando. Inserta en la base de datos un nuevo registro con ese requisito recientemente añadido.

- ***getArtifacts(Integer, Integer)***: Recibe como parámetros el identificador del requisito para el que se quieren obtener los artefactos asociados y el identificador del proyecto en curso. Devuelve una lista de artefactos que tienen asociados el requisito cuyo id se recibe como parámetro.
  - ***getTableName(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos “Requisitos”.
  - ***complete(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto “Requisitos” con los datos obtenidos de la consulta a base de datos.
  - ***update(Object, Integer)***: Recibe como parámetro el requisito que se quiere modificar y el identificador del proyecto al que pertenece. Modifica los datos referidos al requisito que se le indica.
  - ***getReferenceRequirement(Integer, Integer)***: Recibe por parámetro el identificador de un requisito y el proyecto al que pertenece. Devuelve el nombre del “Reference Requirement” al que hace referencia.
- **ProjectDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada proyecto almacenado en la base de datos. Consta de los siguientes métodos:
    - ***getProject()***: Método que devuelve todos los proyectos que existen actualmente almacenados en la base de datos.
    - ***insertProject()***: Método que inserta un nuevo proyecto en la base de datos con los datos que ha introducido el usuario en la aplicación.
    - ***getTableName(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos proyectos.
    - ***complete(ResultSet)***: Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto proyecto con los datos obtenidos de la consulta a base de datos.
  - **RoleDAO**: Clase que contiene todos los métodos encargados de la manipulación de los datos de cada rol almacenado en la base de datos. Consta de los siguientes métodos:

- **countReferences(Integer):** Recibe por parámetro el proyecto con el que está trabajando el usuario. Devuelve el número total de roles que han sido asociados a los *Reference Role* que existen en el proyecto.
  - **insertRole(Roles, Integer):** Recibe por parámetro el rol que ha introducido el usuario en la aplicación y el proyecto con el que se está trabajando. Inserta en la base de datos un nuevo registro con ese rol recientemente añadido.
  - **getArtifacts(Integer, Integer):** Recibe como parámetros el identificador del rol para el que se quieren obtener los artefactos asociados y el identificador del proyecto en curso. Devuelve una lista de artefactos que tienen asociados el rol cuyo id se recibe como parámetro.
  - **getTableName(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan los distintos roles.
  - **complete(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto roles con los datos obtenidos de la consulta a base de datos.
  - **update(Object, Integer):** Recibe como parámetro el rol que se quiere modificar y el identificador del proyecto al que pertenece. Modifica los datos referidos al rol que se le indica.
  - **getReferenceRole(Integer, Integer):** Recibe por parámetro el identificador de un rol y el proyecto al que pertenece. Devuelve el nombre del *Reference Role* al que hace referencia.
- **TechniqueDAO:** Clase que contiene todos los métodos encargados de la manipulación de los datos de cada técnica almacenada en la base de datos. Consta de los siguientes métodos:
    - **countReferences(Integer):** Recibe por parámetro el proyecto con el que está trabajando el usuario. Devuelve el número total de técnicas que han sido asociadas a los *Reference Technique* que existen en el proyecto.
    - **insertTechnique(Tecnicas, Integer):** Recibe por parámetro la técnica que ha introducido el usuario en la aplicación y el proyecto con el que se está trabajando. Inserta en la base de datos un nuevo registro con esa técnica recientemente añadida.
    - **getArtifacts(Integer, Integer):** Recibe como parámetros el identificador de la técnica para la que se quieren obtener los artefactos asociados y el

- identificador del proyecto en curso. Devuelve una lista de artefactos que tienen asociados la técnica cuyo id se recibe como parámetro.
- **getTableNames(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Devuelve el nombre de la tabla de la base de datos donde se almacenan las distintas técnicas.
  - **complete(ResultSet):** Recibe por parámetro el acceso a los resultados de una sentencia SQL. Rellena un nuevo objeto técnicas con los datos obtenidos de la consulta a base de datos.
  - **update(Object, Integer):** Recibe como parámetro la técnica que se quiere modificar y el identificador del proyecto al que pertenece. Modifica los datos referidos a la técnica que se le indica.
  - **getReferenceTechnique(Integer, Integer):** Recibe por parámetro el identificador de una técnica y el proyecto al que pertenece. Devuelve el nombre del *Reference Technique* al que hace referencia.
- **LinkagesDAO:** Clase que contiene los accesos a la base de datos referidos a las tablas intermedias o de relación entre las tablas principales expuestas en los puntos anteriores. Contiene los siguientes métodos:
    - **getAttributes(Integer, Integer):** Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los distintos atributos que, según el modelo, se pueden añadir al artefacto que se quiere manipular, siempre que sean de tipo enumerado.
    - **getAttributesNoEnum(Integer, Integer):** Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los distintos atributos que, según el modelo, se pueden añadir al artefacto que se quiere manipular, salvo aquellos que son de tipo enumerado.
    - **getEvents(Integer, Integer):** Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los distintos eventos que se han añadido para ese artefacto.
    - **getEvidence(Integer, Integer):** Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se

está trabajando. Muestra los requisitos que están asociados con ese artefacto concreto.

- ***getNoEvidence(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los requisitos que aún no están asociados con ese artefacto concreto, pero que cumplen las características para que puedan estarlo.
- ***getNoParticipant(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los roles o participantes que aún no están asociados con ese artefacto, pero que según el modelo de referencia podrían estarlo.
- ***getNotTechniqueUse(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra las técnicas que aún no están asociadas con ese artefacto concreto, pero que podrían estarlo según lo estipulado en el modelo que se está usando como referencia.
- ***getParticipant(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los roles o participantes que están asociados con ese artefacto.
- ***getProperties(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra las propiedades añadidas al artefacto concreto que se está viendo.
- ***getRelationships(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra las distintas relaciones existentes para ese artefacto concreto.
- ***getRequirement(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra los requisitos que se pueden asociar con el artefacto.
- ***getTechniqueUse(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto con el que se está trabajando. Muestra las técnicas que están asociadas con ese artefacto concreto.



- ***insertArtifactResponsible(Artifact, Role, Integer)***: Recibe por parámetros el artefacto y el rol que se quieren asociar, y el proyecto en el que se está trabajando. Inserta en la base de datos la nueva asociación entre ambos elementos.
- ***insertEvaluation(Evaluation, Integer)***: Recibe por parámetros la evaluación a añadir y el proyecto en el que se está trabajando. Inserta en la base de datos la nueva evaluación de un artefacto concreto.
- ***insertEvent(Event, Integer)***: Recibe por parámetros el evento a añadir y el proyecto en el que se está trabajando. Inserta en la base de datos el nuevo evento de un artefacto concreto.
- ***insertProperty(Artifact, Integer, Integer, String)***: Recibe por parámetros el artefacto con el que se está trabajando, el atributo que se le quiere añadir, su valor y el proyecto en el que se está trabajando. Inserta en la base de datos la nueva característica del artefacto.
- ***insertRelationship(Artifact, Integer, Integer, Integer, String)***: Recibe por parámetros el artefacto al que se le quiere añadir una relación, el identificador del artefacto con el que se le quiere relacionar, el identificador de la referencia de esa relación, el proyecto en el que se está trabajando y el nombre de la relación. Inserta en la base de datos la nueva relación entre los dos artefactos que se quieren asociar.
- ***insertRequirement(Artifact, Requirement, Integer)***: Recibe por parámetros el artefacto y el requisito que se quieren asociar, y el proyecto en el que se está trabajando. Inserta en la base de datos la nueva asociación entre ambos elementos.
- ***insertTechniqueUse(Artifact, Technique, Integer)***: Recibe por parámetros el artefacto y la técnica que se quieren asociar, y el proyecto en el que se está trabajando. Inserta en la base de datos la nueva asociación entre ambos elementos.
- ***removeArtifactResponsible(Artifact, Integer)***: Recibe por parámetro el artefacto con el que se está trabajando y el proyecto en curso. Borra las asociaciones entre los roles y el artefacto en cuestión.
- ***removeEvidence(Artifact, Integer)***: Recibe por parámetro el artefacto con el que se está trabajando y el proyecto en curso. Borra las asociaciones entre los requisitos y el artefacto en cuestión.

- ***removeTechniqueUse(Artifact, Integer)***: Recibe por parámetro el artefacto con el que se está trabajando y el proyecto en curso. Borra las asociaciones entre las técnicas y el artefacto en cuestión.
- ***showAttributes(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto en el que se está trabajando. Devuelve una lista de atributos asociados a ese artefacto según el modelo.
- ***showRelationships(Integer, Integer)***: Recibe por parámetro el identificador del artefacto que se está manipulando y el identificador del proyecto en el que se está trabajando. Devuelve las relaciones que tiene ese artefacto según el modelo.

#### 4.2.6 Diagramas de secuencia

A continuación se detalla el diagrama de secuencia de las acciones llevadas a cabo durante el uso de la aplicación.

Debido a la gran cantidad de acciones que se llevan a cabo en el total de la aplicación, se ha decidido mostrar los diagramas de secuencia referidos a cada caso de uso de los expuestos.

##### 4.2.6.1 Caso de Uso: Crear Proyecto

El usuario deberá dirigirse a la pantalla de configuración de la herramienta, y añadir los datos necesarios para crear el proyecto. El programa analizará el modelo introducido y lo almacenará en base de datos, y creará el nuevo proyecto.

El diagrama de secuencia es por tanto el siguiente: El responsable del proyecto, a través de la vista, se comunica a través de la llamada *doPost()* con el servlet que va a realizar la conexión con la parte del modelo de la arquitectura. El servlet coge los datos que le ha enviado el usuario y los envía a la clase *ReadFile*, donde se produce el análisis del fichero que se ha subido a la aplicación como modelo de referencia. Una vez que está analizado y organizado, esta clase llama a los métodos de la clase *ProjectDAO*, a través de los cuales se almacena la información en la base de datos. Una vez hecho, dicha clase devuelve los objetos creados al servlet, y este envía a la vista los nuevos datos, y la vista del nuevo proyecto por pantalla, al usuario.

Con esto, el esquema quedaría de la siguiente forma:

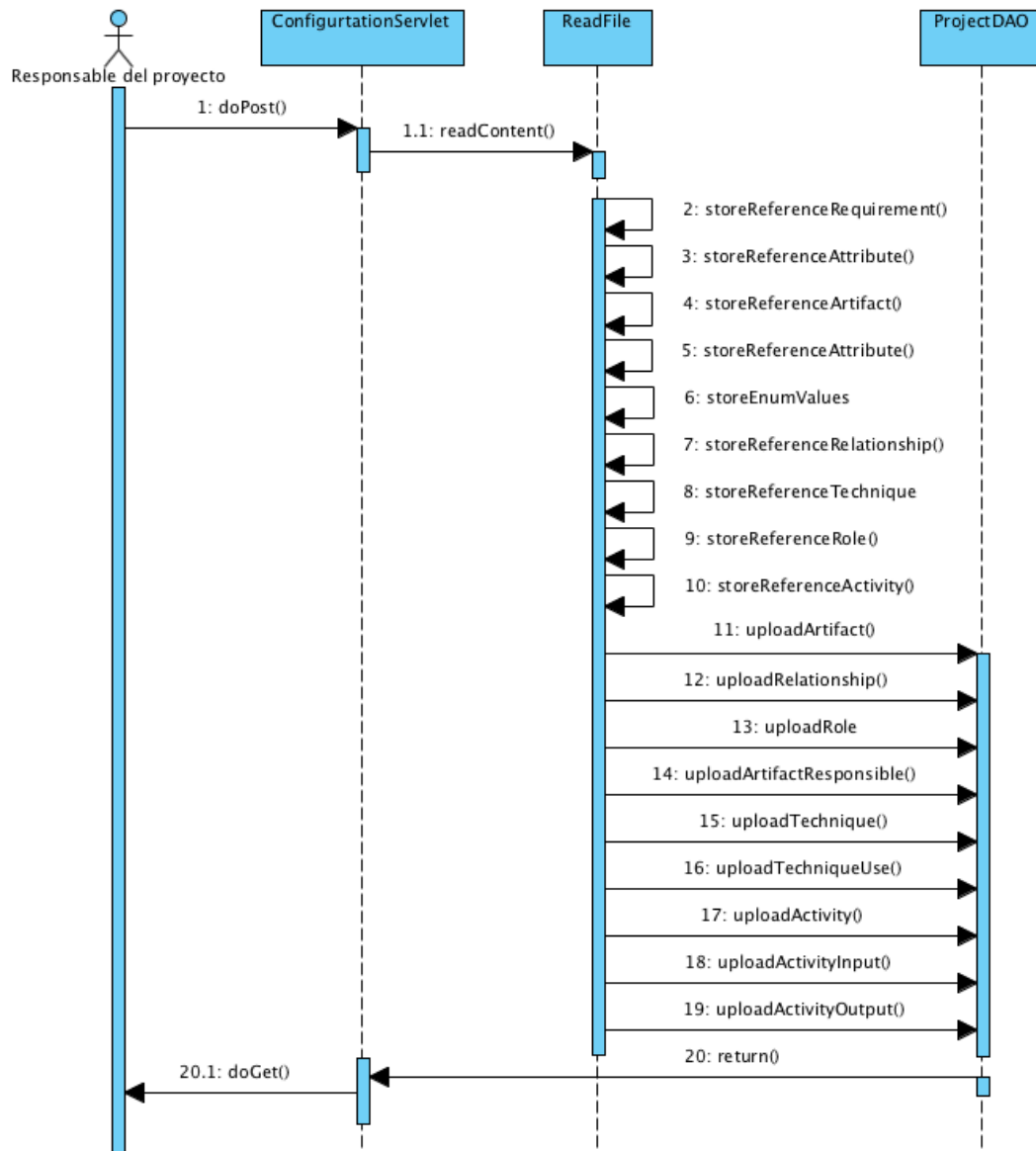


Ilustración 39: Diagrama de secuencia del caso de uso "Crear Proyecto".

#### 4.2.6.2 Caso de Uso: Monitorizar Progreso

El usuario, una vez seleccionado el proyecto, deberá dirigirse a la pantalla de progreso, desde la cual se verá el avance o la completitud del proyecto.

El responsable entra en la página de visualización del progreso del proyecto, comunicándose a través de ella directamente con el servlet correspondiente. Desde esta clase, el programa se comunica con todas las clases DAO de los elementos que se van a visualizar a través de la creación de objetos de la misma. Para obtener la cantidad total de elementos

existentes de referencia, se usará el método `getCount()` y, para los elementos introducidos por el usuario, el método `countReferences()`. Estos métodos devuelven un número de tipo entero al servlet, que se los mostrará por pantalla al usuario.

Debido a la cantidad de clases que intervienen en este caso de uso, se ha dividido el diagrama en dos para facilitar su legibilidad. Los esquemas, por lo tanto, serán los siguientes:

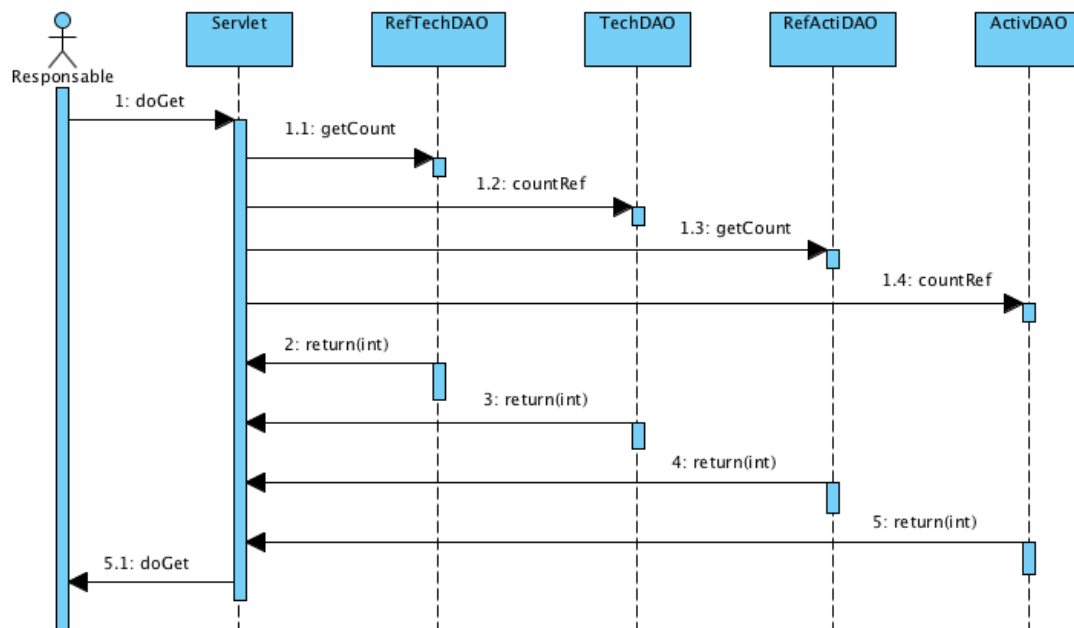


Ilustración 40: Diagrama de secuencia del caso de uso "Monitorizar Progreso" Primera parte.

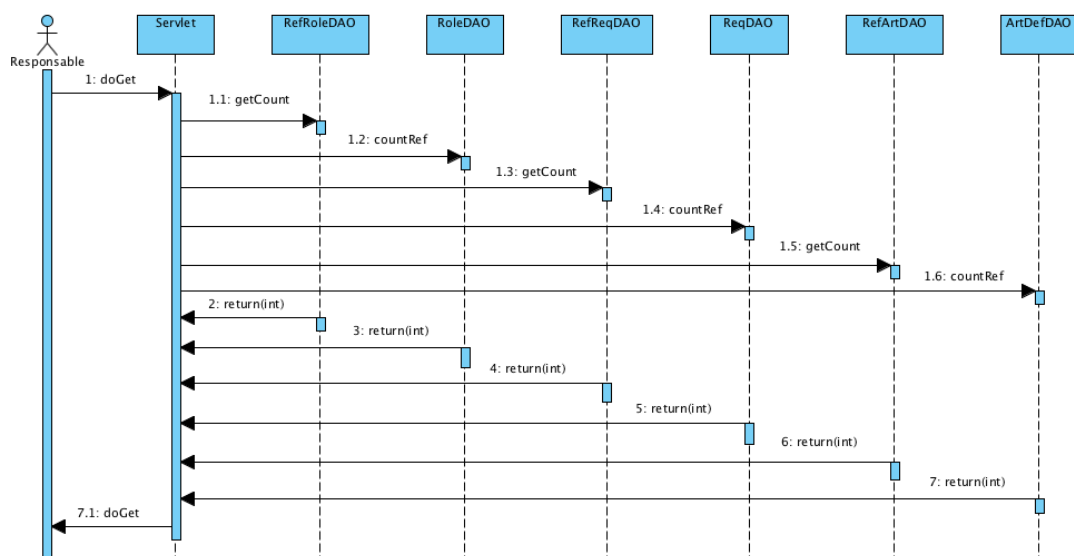


Ilustración 41: Diagrama de secuencia del caso de uso "Monitorizar Progreso" Segunda Parte.

#### 4.2.6.3 Caso de Uso: Gestionar Artefactos

El usuario se introduce en la página principal de los artefactos. Desde ella, crea un *artifact definition*. Una vez creado, creará un nuevo artefacto al que, seguidamente, se le asociaran el resto de elementos.

El responsable de aseguramiento crea un nuevo *Artifact Definition*, para esto, el servlet se comunica con *ArtifactDefinitionDAO* a través de la llamada *insertArtifactDefinition()*, que le devuelve el objeto creado. Una vez que está creado y añadido, el responsable crea un nuevo artefacto relacionado con ese *artifactDefinition*. Para ello, se comunica con la clase *ArtifactDAO* a través del método *insertArtifact()*. Además, enlaza con la clase *LinkagesDAO* para insertar las técnicas y roles que tenía su artefacto anterior, a través de los métodos *insertArtifactResponsible* e *inserTechniqueUse*, que devuelven al servlet los nuevos objetos creados. Una vez creado, el servlet envía a la vista la nueva interfaz con esos artefactos introducidos. Para añadir las correspondientes asociaciones, el usuario, desde la vista, vuelve a mandar una petición al servlet. Este, cuando la recibe, se comunica con la clase *LinkagesDAO* de nuevo para añadir las asociaciones que el usuario decida añadir. Todas estas devuelven al servlet el objeto introducido, y esta devuelve al usuario la vista con las asociaciones añadidas.

El esquema de esta secuencia sería, por lo tanto, el siguiente:

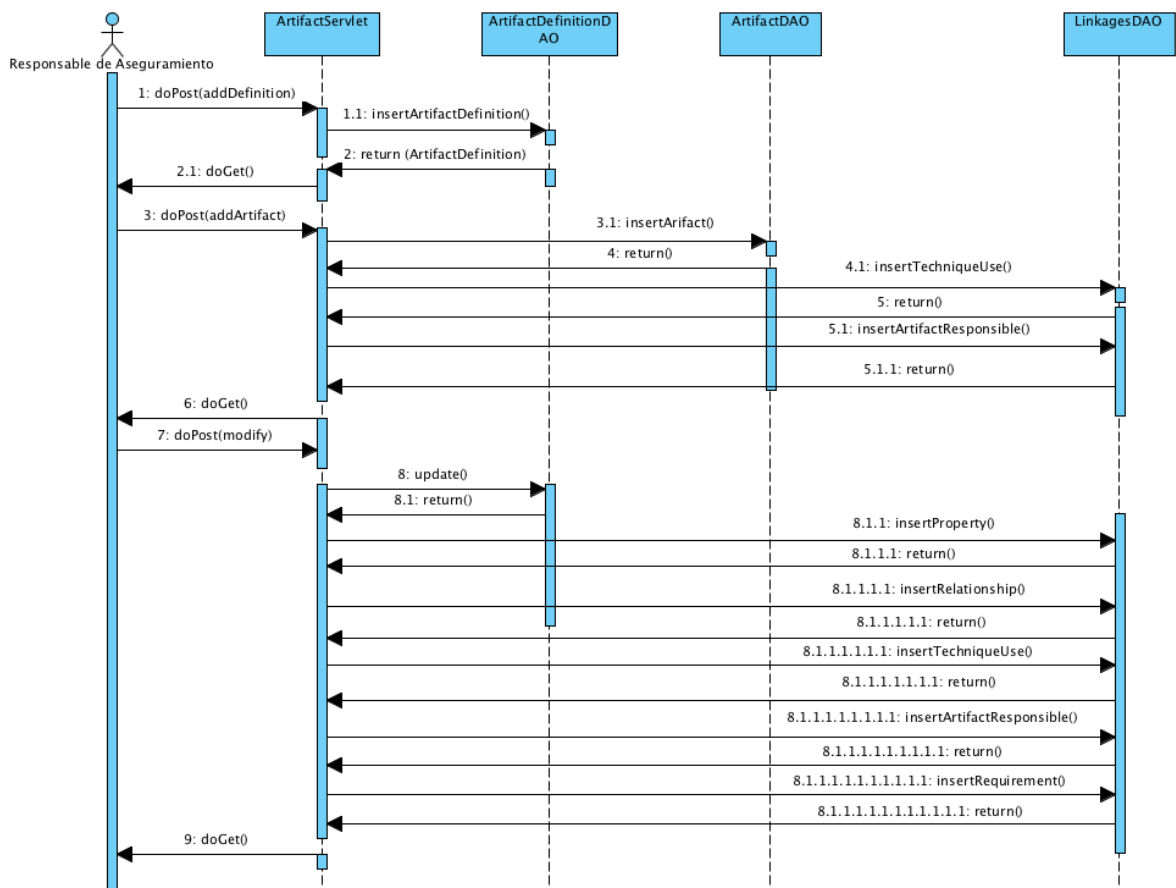


Ilustración 42: Diagrama de secuencia del caso de uso "Gestionar Artefactos".

#### 4.2.6.4 Caso de Uso: Gestionar Información de procesos

El usuario accede a la pantalla de procesos y añade nuevas técnicas, roles y actividades al proyecto. En el caso de la actividad, una vez creada, añadirá las asociaciones correspondientes con los artefactos que ya existen para el proyecto.

El responsable de aseguramiento, entra en la pantalla de procesos y añade una nueva técnica. La vista envía la petición al servlet correspondiente, *TechniqueServlet*, y este, a través de un objeto de la clase *TechniqueDAO*, inserta la nueva técnica en el sistema mediante el método *insertTechnique()*. Una vez insertado, dicho método devuelve al servlet el objeto creado, y este se lo reenvía a la vista, desde donde el responsable podrá ver como se ha añadido al proyecto.

Una vez creada la técnica, el responsable de aseguramiento añade un nuevo participante al proyecto. Para ello, la vista correspondiente reenvía la petición al servlet, en

este caso *RoleServlet*. Esta clase, a través de un objeto de la clase *RoleDAO*, inserta en la base de datos el nuevo participante. Esta clase devuelve al servlet el objeto que ha creado, y este lo reenvía a la vista para que el usuario pueda acceder a él.

En el caso de añadir una nueva actividad, el proceso es el mismo que en los anteriores. La vista envía la petición del usuario al servlet, y este se comunica con la clase *ActivityDAO*, desde donde se insertará en la base de datos la nueva actividad. Dicha clase devuelve al servlet el objeto creado, y este reenvía la información a la vista para que el responsable pueda visualizarla. Una vez añadida la actividad, para poder insertar las correspondientes asociaciones con los artefactos, el usuario vuelve a enviar una petición al servlet, en este caso de modificación. El servlet se volverá a comunicar con la clase *ActivityDAO*, desde donde se harán dichas inserciones mediante los métodos *insertActivityInput()* e *insertActivityOutput()*. La clase devuelve al servlet los objetos creados, y este los reenvía a la vista.

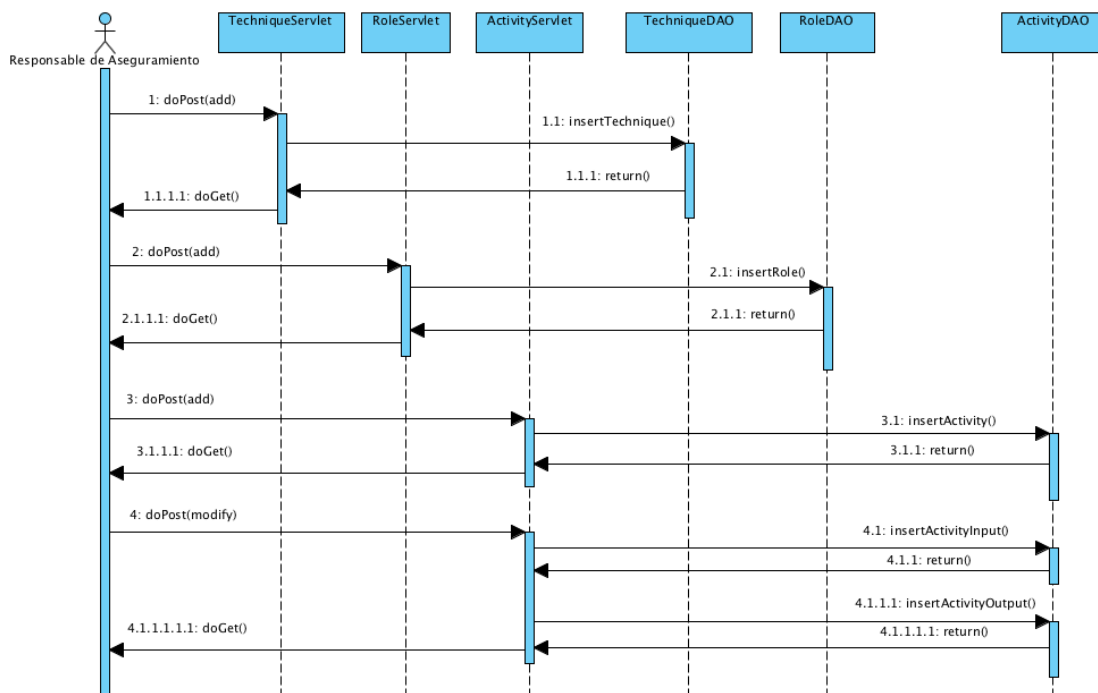


Ilustración 43: Diagrama de secuencia "Gestionar informacion de proceso".

#### 4.2.6.5 Caso de Uso: Gestionar Información de Justificación

El usuario accede a la pantalla de justificación de la aplicación e introduce un nuevo requisito a su requisito de referencia correspondiente.

El responsable de aseguramiento, a través de la vista, manda una petición al servlet, en este caso *RequirementServlet*, para poder añadir un nuevo requisito. El servlet se comunica con la clase *RequirementDAO*, desde donde se hará la inserción en la base de datos mediante el método *insertRequirement()* de la misma. Una vez añadido, esta clase devuelve al servlet el objeto recientemente creado, y este renvía toda la información sobre los requisitos que hay en el sistema a la vista, de forma que el usuario pueda trabajar con esa información.

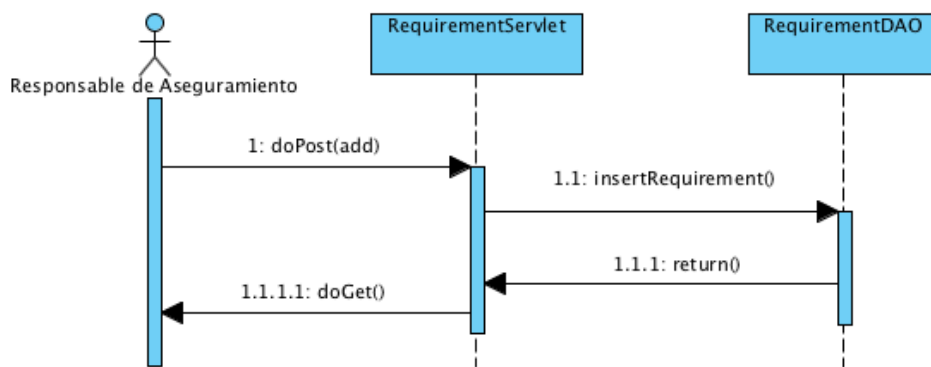


Ilustración 44: Diagrama de secuencia del caso de uso "Gestionar información de justificación".



## 5. Uso de la Aplicación

En este apartado se explicará de forma breve el funcionamiento de la herramienta. Se comenzará explicando los pasos previos necesarios para poder utilizarla, para continuar con un breve resumen de su uso.

### 5.1 Creación del Reference Assurance Framework

A continuación se van a detallar los pasos de un hipotético caso de creación de un nuevo modelo, que más adelante se utilizará para gestionar un proyecto concreto.

- **Paso 0. Creación de un nuevo fichero `model.raf_diagram`:** El usuario accede al *plugin* creado en Eclipse y creará un nuevo fichero de tipo `.Raf_Diagram`. Al crear este, se creará a la vez un fichero paralelo con la extensión `.raf`.

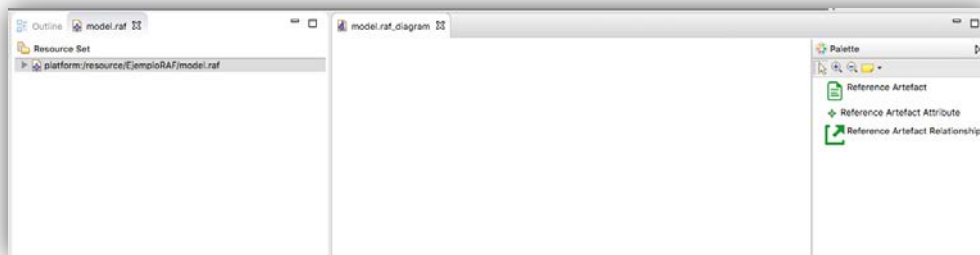


Ilustración 45: Pantalla inicial del editor.

- **Paso 1. Creación de un nuevo *Reference Artefact*:** El usuario arrastra a la pantalla principal un nuevo *Reference Artefact*. Al hacerlo, aparecerá en la parte de debajo de la pantalla un cuadro de propiedades desde el cual se añadirán los datos básicos del elemento. Al añadirlo desde el editor gráfico, automáticamente quedará añadido también en el árbol de la sección izquierda. La apariencia de la herramienta en este caso es la siguiente:

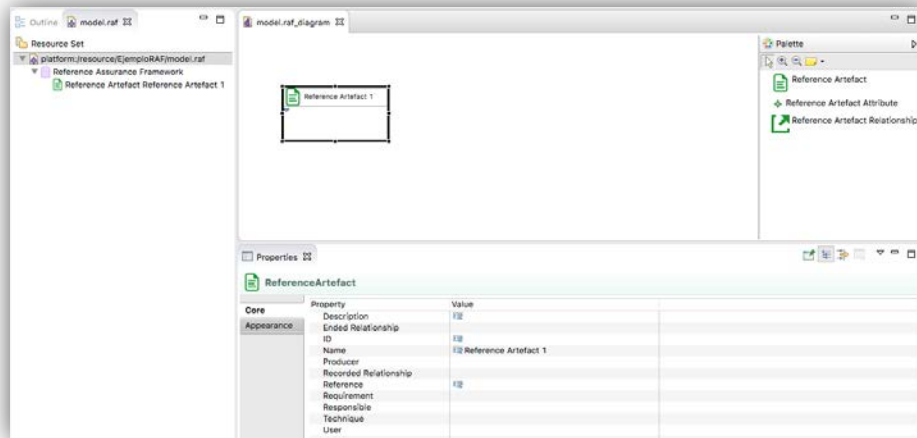


Ilustración 46: Añadir un nuevo Reference Artefact.

- Paso 2. Añadir atributos a un *Reference Artefact*:** Arrastrando desde la sección derecha de la página un nuevo atributo se añade al artefacto. Una vez añadido, se podrán modificar sus datos en la sección de debajo de propiedades y, en la parte izquierda de la misma, se añade ese mismo atributo al artefacto en cuestión.

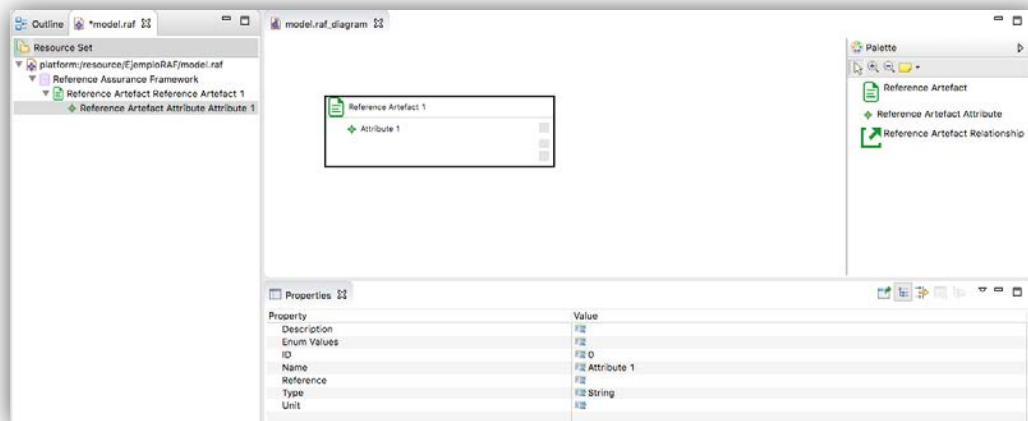


Ilustración 47: Añadir un nuevo atributo.

- Paso 3. Añadir una relación entre artefactos:** Una vez creado un *Reference Artefact*, se podrá añadir una relación con otro. Esto se hace de la misma forma que en los casos anteriores, arrastrando el elemento “*Reference Artefact Relationship*” de la sección izquierda a la pantalla principal. Dicha relación se

añadirá al diagrama y al árbol que se muestra en la figura de la sección izquierda. El diagrama quedaría de la siguiente forma:

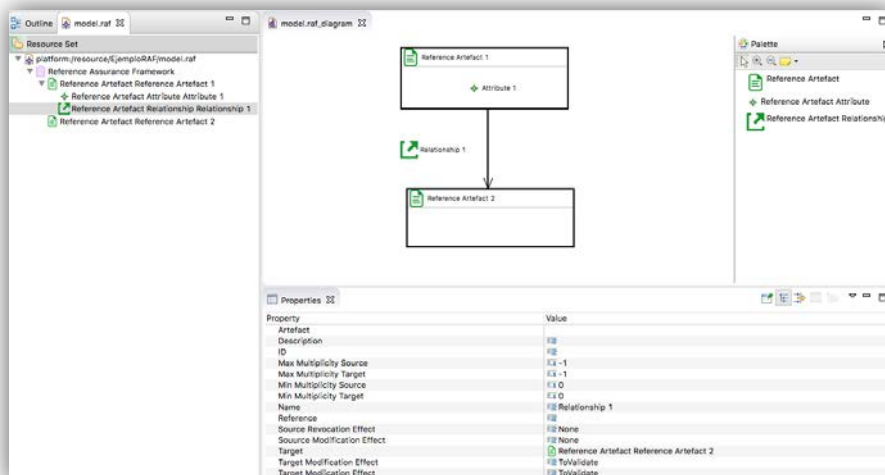


Ilustración 48: Añadir una nueva relación entre artefactos.

- Paso 4.** Creación de una nueva técnica: Los elementos distintos a los artefactos, los atributos y las relaciones se crean directamente desde el árbol de la sección izquierda. Para crear una nueva técnica, se selecciona el *Reference Assurance Framework* y se le crea un nuevo hijo de tipo *Reference Technique*. Una vez creado, en la sección inferior se podrán añadir las propiedades correspondientes a la misma. Se añadirán también así las relaciones con los artefactos.

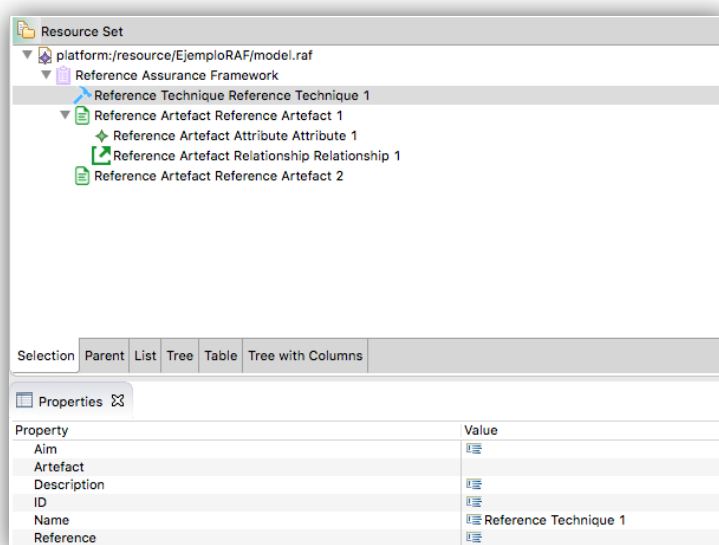


Ilustración 49: Creación de una nueva técnica.

- **Paso 5. Creación de un nuevo rol:** Al igual que las técnicas, los roles se crean desde la sección izquierda. Para crearlo, se selecciona el *Reference Assurance Framework* y se le crea un nuevo hijo de tipo *Reference Role*. Una vez creado, en la sección inferior se podrán añadir las propiedades correspondientes al mismo. Se añadirán también así las relaciones con los artefactos.

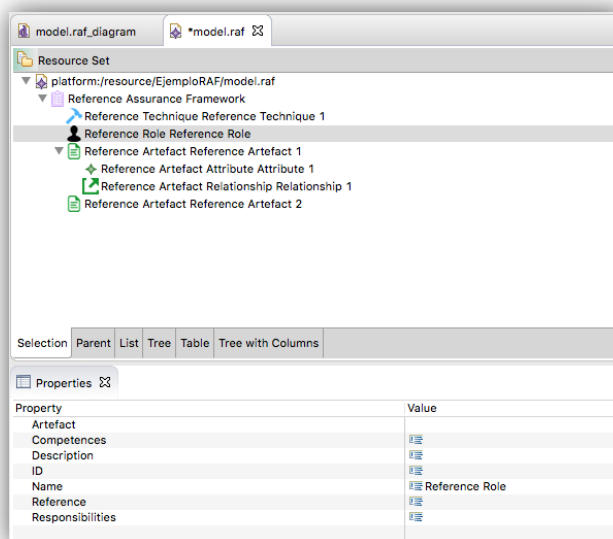


Ilustración 50: Creación de un nuevo rol.

- **Paso 6. Creación de una nueva actividad:** Como en los casos anteriores, se crea desde el menú izquierdo. Para crearlo, se selecciona el *Reference Assurance Framework* y se le crea un nuevo hijo de tipo *Reference Activity*. Una vez creado, en la sección inferior se podrán añadir las propiedades correspondientes a la actividad en cuestión. Se añadirán también así las relaciones con los artefactos.

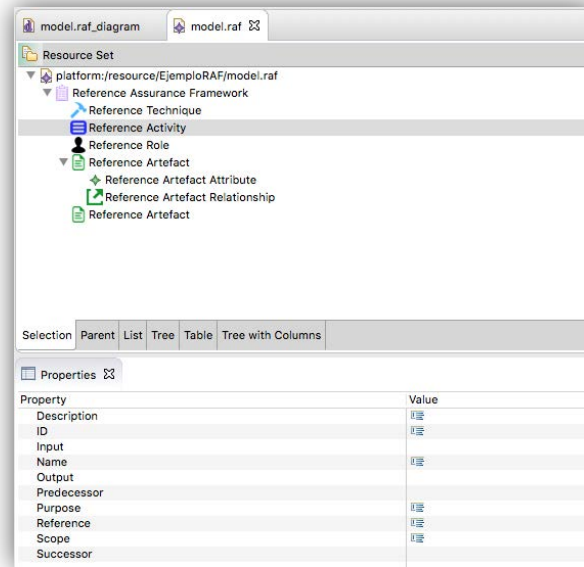


Ilustración 51: Creación de una nueva actividad.

- Paso 7. Creación de un nuevo requisito:** Se selecciona el *Reference Assurance Framework* y se le crea un nuevo hijo de tipo *Reference Requirement*. Una vez creado, en la sección inferior se podrán añadir las propiedades correspondientes a dicho requisito. Se añadirán también así las relaciones con los artefactos.

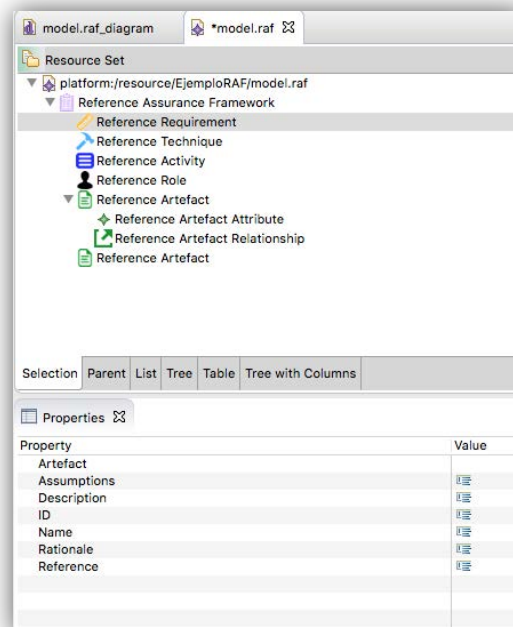


Ilustración 52: Creación de un nuevo requisito.

- **Paso 8. Finalización:** Una vez añadidos todos los elementos que son necesarios según el estándar que se está teniendo en cuenta, se creará un fichero tipo .raf, que contiene un archivo con formato XML, que será el que se deberá añadir a la aplicación a la hora de crear un nuevo proyecto.

## 5.2 Requisitos de instalación previos al uso de la aplicación

Antes de poner en marcha la herramienta, existen ciertos aspectos a tener en cuenta, y sin los cuales la aplicación no funcionará correctamente.

- **Compilador:** Por si pudieran producirse problemas de compatibilidad, se recomienda descargar e instalar la librería jdk1.8 de Java. Si se usase alguna versión anterior, podría darse el caso de que haya ciertas funciones que no funcionasen de forma correcta.
- **Servidor:** La herramienta implementada está alojada en Tomcat8. Para que la aplicación se pueda conectar correctamente a la base de datos, será necesario añadir a la carpeta */lib* del mismo el conector.
- **Base de datos:** La base de datos usada en este caso es *MySQL*. El esquema necesario para su correcto funcionamiento se ha expuesto en el apartado [4. Diseño e Implementación](#), y su script correspondiente se encuentra dentro de la carpeta *Database* del proyecto.

### 5.2.1 Archivos de configuración

La herramienta está pensada para que pueda ser utilizada por cualquier usuario. Por ello, se han creado dos archivos de configuración que se encuentran en la carpeta raíz del proyecto. Estos son:

- **Configuration.properties:** En él habrá que detallar la cadena de conexión a la base de datos, el usuario y la contraseña y la ruta en la que se van a generar los archivos del modelo.
- **Log4j.properties:** Archivo de configuración del *log* de la aplicación. En él habrá que indicar el detalle con el que se quiere obtener y la ruta en la que va a guardarse dicho archivo.

El aspecto del fichero de configuración **configuration.properties** es el siguiente:

name	value
database_user	
database_password	
update_file	
database_url_conection	

Ilustración 53: Archivo *configuration.properties*

El aspecto del fichero de configuración **log4j.properties** es el siguiente:

name	value
log4j.rootLogger	TRACE, file, stdout
log4j.appender.file	org.apache.log4j.RollingFileAppender
log4j.appender.file.File	
log4j.appender.file.MaxFileSize	10MB
log4j.appender.file.MaxBackupIndex	10
log4j.appender.file.layout	org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern	%d{yyyy-MM-dd HH:mm:ss} %-5p-%m%n
log4j.appender.stdout	org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target	System.out
log4j.appender.stdout.layout	org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern	%d{yyyy-MM-dd HH:mm:ss} %-5p-%m%n

Ilustración 54: Archivo *log4j.properties*.

## 5.3 Guía de uso de la aplicación

A continuación se va a describir una posible forma de uso de la aplicación.

- **Paso 0. Desplegando la aplicación:** En el momento en el que la aplicación ha sido puesta en marcha, se abrirá una ventana en la que se ven las siguientes opciones:
  - *Project:* Avance del proyecto que se está llevando a cabo. En esta pestaña aparecen los porcentajes y términos absolutos de los distintos elementos concretos del proyecto con relación al modelo que se está usando como referencia.
  - *Artifacts:* Artefactos concretos añadidos al proyecto. En esta pestaña e muestran todos los añadidos al proyecto, ordenados según el artefacto de referencia correspondiente. A través de esta opción se podrán visualizar

también las distintas relaciones existentes con el resto de elementos del proyecto. Desde esta opción se podrán llevar a cabo todas las acciones de adición, modificación y borrado de artefactos.

- *Processes*: Técnicas, roles y actividades del proyecto. Accediendo a esta pestaña se podrán crear, modificar y borrar los elementos anteriormente citados.
- *Justification*: Requisitos asociados al proyecto en curso. En este caso se podrán visualizar los distintos elementos añadidos en función de los requisitos de referencia del modelo elegido. En este caso, se podrán llevar a cabo todas las modificaciones que tengan relación con los requisitos específicos.
- *Configuration*: Pantalla en la que se selecciona un proyecto o bien se crea uno nuevo. En caso de que no se seleccione ninguno, o no se cree, el resto de pantallas de la aplicación aparecerán vacías.



Ilustración 55: Pantalla inicial.

- **Paso 1. Creación de un nuevo proyecto:** A partir de la opción “*Configuration*” del menú principal se accede a la pantalla que se muestra a continuación. Para crear un nuevo proyecto, habrá que insertar el nombre, una descripción (opcional) y el modelo al que va a hacer referencia el proyecto. Estos datos son definitivos y en ningún caso podrán modificarse. Una vez añadido el proyecto, aparecerá en el menú de la izquierda. Para poder trabajar con él, lo único que habrá que hacer es seleccionarlo.



RAF

Home Project Artifacts Process Justification

### Configuration

**Choose an existing Project**

Example Project

**Create a new Project**

Name*	
Description	
Model*	Seleccionar archivo Ningún a...ccionado

Submit

Trabajo Fin de Grado 2016. Elena Correias Montiel

Ilustración 56: Pantalla de configuración.

- **Paso 2. Creación de un nuevo “Artifact Definition”:** Una vez seleccionado el proyecto, dentro del menú “Artifacts” se verán todos los artefactos de referencia del modelo que se ha asociado a dicho proyecto. Para añadir un nuevo “Artifact Definition”, habrá que pulsar sobre el botón “Add a new Artifact Definition”. Se abrirá una nueva pestaña en la que habrá que indicar el nombre y la descripción del mismo.

RAF

Home Project Artifacts Processes Justification

### Add a new Artifact Definition

Name*	
Description*	

Submit Cancel

Trabajo Fin de Grado 2016. Elena Correias Montiel

Ilustración 57: Pantalla para introducir un nuevo Artifact Definition

- **Paso 3. Creación de un nuevo “Artifact” asociado a un “Artifact Definition”:** Una vez creado un “Artifact Definition”, se le podrán añadir versiones del mismo. Para ello, habrá que entrar en la opción “See Artifacts”, y pulsar en el botón “Add a new Artifact”. Esto llevará al usuario a una nueva ventana en la que únicamente podrá introducir el nombre del artefacto, su descripción y la versión del mismo. El resto de componentes (asociaciones, etc.) se llevarán a cabo a través de la opción de editar el mismo.

The screenshot shows the RAF application interface. At the top is a dark blue header with the 'RAF' logo in white. Below the header is a navigation bar with links: Home, Project, Artifacts, Process, and Justification. The main content area has a title 'Add a new Artifact of Artifact Definition 1'. Below the title is a form with four rows: 'Name\*' (with a red asterisk), 'Description', 'Version', and 'Previous Version\*' (with a red asterisk). Each row has a text input field. The 'Previous Version\*' field has a small dropdown arrow on the right. Below the form are two buttons: 'Submit' and 'Cancel'. At the bottom of the page is a dark blue footer with the text 'Trabajo Fin de Grado 2016. Elena Correás Montiel'.

Name*	
Description	
Version	
Previous Version*	

Submit Cancel

Trabajo Fin de Grado 2016. Elena Correás Montiel

Ilustración 58: Pantalla para introducir un nuevo artefacto.

- **Paso 4. Creación de una nueva técnica:** A través de la opción “Processes” se accede a la parte de la aplicación encargada de administrar los distintos elementos referidos a los procesos. Estos son: técnicas, roles y actividades. Para crear una nueva técnica, dentro del menú “Techniques”, sabiendo cuál es la “ReferenceTechnique” a la que se le quiere añadir una nueva técnica, habrá que acceder a la opción “Add a new Technique”. Una vez ahí, habrá que añadir el nombre y la descripción de la técnica. Dándole a “Submit” ya estaría la técnica creada. Si el usuario quiere modificar los datos de alguna técnica, deberá entrar en la opción “Edit” de la misma, y añadir los nuevos campos.

The screenshot shows the RAF web application interface. At the top is a dark blue header with the 'RAF' logo in white. Below the header is a navigation bar with links: Home, Project, Artifacts, Processes, and Justification. The main heading is 'Processes: Add a new Technique'. On the left, there is a sidebar with three options: TECHNIQUES, ROLES, and ACTIVITIES. The main content area contains a form with two input fields: 'Name\*' and 'Description\*'. Below the form are two buttons: 'Submit' and 'Cancel'. At the bottom of the page, there is a footer that reads 'Trabajo Fin de Grado 2016. Elena Correas Montiel'.

Ilustración 59: Pantalla para introducir una nueva técnica.

- **Paso 5. Creación de un nuevo rol:** Accediendo de la misma forma que para el caso de las técnicas, si lo que el usuario desea es crear un nuevo rol o responsable, deberá, dentro del *Reference Role* al que desee añadirlo, pulsar el botón “Add a new Role”. Se redirigirá al usuario a una nueva pestaña en la que deberá introducir el nombre y la descripción del mismo. Si el usuario quiere modificar los datos de algún rol, deberá entrar en la opción “Edit” del mismo, y añadir los nuevos campos.

The screenshot shows the RAF web application interface. At the top is a dark blue header with the 'RAF' logo in white. Below the header is a navigation bar with links: Home, Project, Artifacts, Processes, and Justification. The main heading is 'Processes: Add a new Rol'. On the left, there is a sidebar with three options: TECHNIQUES, ROLES, and ACTIVITIES. The main content area contains a form with two input fields: 'Name\*' and 'Description\*'. Below the form are two buttons: 'Submit' and 'Cancel'. At the bottom of the page, there is a footer that reads 'Trabajo Fin de Grado 2016. Elena Correas Montiel'.

Ilustración 60: Pantalla para introducir un nuevo rol.

- **Paso 6. Creación de una nueva actividad:** La forma de añadir actividades será la misma que los dos casos anteriores. Será necesario, también, introducir el nombre, la descripción y fechas de comienzo y fin de la misma.

The screenshot shows the RAF web application interface. At the top is a dark blue header with the 'RAF' logo in white. Below the header is a navigation bar with links: Home, Project, Artifacts, Process, and Justification. The main content area is titled 'Process: Add a new Activity'. On the left, there are labels for 'TECHNIQUES', 'PARTICIPANTS', and 'ACTIVITIES'. The 'ACTIVITIES' section contains a form with the following fields:

Name*	
Description	
Start Date	01/01/2013 12:00:00
End Date	01/01/2013 12:00:00

Below the form are two buttons: 'Submit' and 'Cancel'. At the bottom of the page, a dark blue footer contains the text 'Trabajo Fin de Grado 2016. Elena Correás Montiel'.

Ilustración 61: Pantalla para introducir una nueva actividad.

- **Paso 7. Asociar actividades y artefactos:** Una vez estén creados los artefactos y actividades que se quieren relacionar, a través de la opción “Edit” de cada actividad se podrán indicar cuales son los usuarios y los productores de cada artefacto. Para que un artefacto pueda asociarse a una actividad, debe estar así indicado en el modelo que se ha tenido en cuenta, a través de sus correspondientes *Reference Artefacts* y *Reference Activities*. En el caso de que esa asociación pueda existir, se le dará al usuario una lista con todos los artefactos con los que puede hilar una determinada actividad. En caso de que no se pueda, la lista aparecerá vacía. Desde esta misma pestaña el usuario puede modificar los datos añadidos anteriormente.

TECHNIQUES	Name*	Activity 1
PARTICIPANTS	Description	Description of Activity
ACTIVITIES	Start Date	07/06/2016 12:00:00
	End Date	07/06/2016 12:00:00
	Input Artifacts	<input type="checkbox"/> Artifact 1
	Output Artifacts	<input type="checkbox"/> Artifact 1

Trabajo Fin de Grado 2016. Elena Correás Montiel

Ilustración 62: Pantalla de modificación de las actividades.

- Paso 8. Crear un nuevo requisito:** A través de la opción “Justification” se accede a la parte de la aplicación encargada de administrar los requisitos. Para crear un nuevo requisito, sabiendo cuál es el “ReferenceRequirement” al que se le quiere añadir un nuevo elemento, habrá que acceder a la opción “Add a new Requirement”. Una vez ahí, habrá que añadir el nombre y la descripción del requisito. Aceptando el formulario, estaría el requisito creado. Si el usuario quiere modificar los datos de algún requisito, deberá entrar en la opción “Edit” del mismo, y añadir los nuevos campos.

Name*	
Description*	

Trabajo Fin de Grado 2016. Elena Correás Montiel

Ilustración 63: Pantalla para introducir un nuevo requisito.

- **Paso 9. Añadir técnicas, roles, atributos y requisitos a los artefactos:** Una vez añadidos los distintos elementos que forman el modelo, se pueden asociar los artefactos a los mismos. Esto se hace a través de la opción “Edit” de cada artefacto.

RAF	
Home Project Artifacts Process Justification	
Edit "Artifact 1"	
Name*	Artifact 1
Description	
Version	1
Relationships	
Techniques	<input type="checkbox"/> Technique 1 <input type="checkbox"/> Technique 2
Responsibles	<input type="checkbox"/> Participant 1 <input type="checkbox"/> Participant 2
Requirements	<input type="checkbox"/> Requirement 1 <input type="checkbox"/> Requirement 2
Attributes	Reference Attribute 4
	Reference Attribute 5
	Reference Attribute Enumeration 1
	Reference Attribute Enumeration 2
	Reference Attribute Enumeration 3
Location	null
Format	null
<input type="button" value="Submit"/> <input type="button" value="Cancel"/> <input type="button" value="Add Event"/>	

Ilustración 64: Pantalla de modificación de los artefactos.

- **Paso 10. Añadir un evento a un artefacto:** A raíz de la pantalla anterior, se puede añadir un nuevo evento al artefacto en cuestión. Estos eventos pueden ser de cuatro tipos: creación, evaluación, modificación y revocación. Una vez añadido el evento que se necesite, pulsando “Submit” quedará almacenado y se verá en la opción “See Artifact” del artefacto al que se le ha añadido. En este ejemplo, el evento que se va a añadir es de tipo evaluación.

**RAF**

Home Project Artifacts Process Justification

**Add an Event for Artifact 1**

Event type*	Evaluation
Event	evaluation
Responsible*	
Name*	
Description*	
Time*	01/01/2013 12:00:00
Criterion*	
Criterion Description*	
Result*	
Rationale*	

Submit Back

Trabajo Fin de Grado 2016, Elena Correias Montiel

Ilustración 65: Pantalla para añadir un evento.

- **Paso 11.** Ver progreso del proyecto: A través de la opción “Project” del menú, se accede a la pantalla que indica el progreso del proyecto en curso. Esta ventana es de sólo lectura, no pudiendo cambiar ninguno de los datos que aparecen en ella.

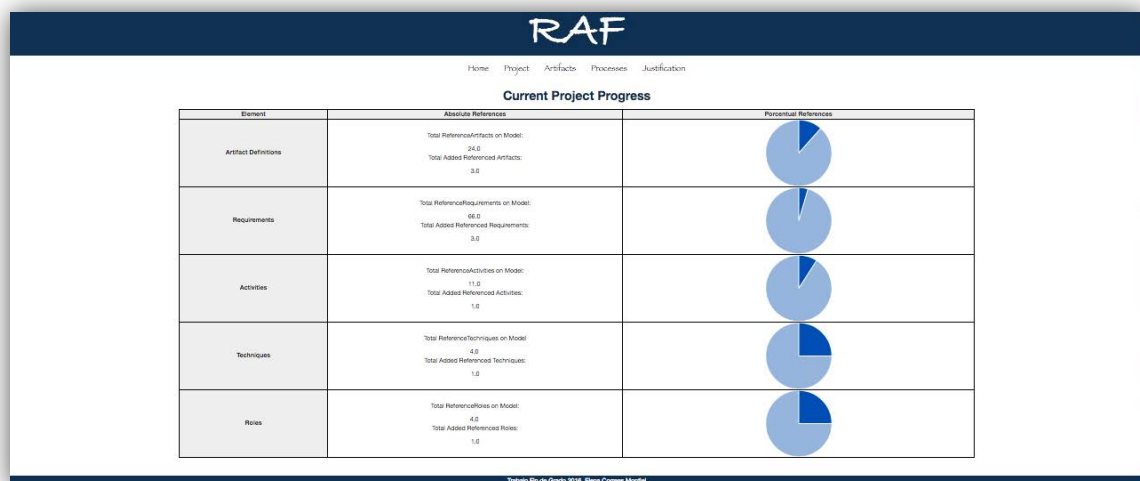


Ilustración 66: Pantalla para ver el progreso del proyecto.

## 6. Pruebas de Sistema

En este apartado se explican las distintas pruebas que se han llevado a cabo para la comprobación de la herramienta. Una vez expuestas, con sus resultados, se añade una matriz de trazabilidad que corrobora que se han probado todos los requisitos y, por lo tanto, todos los casos de uso, explicados durante el documento.

### 6.1 Formato de las pruebas

El formato con el que se expondrán las distintas pruebas que se han realizado es el siguiente:

<b>Identificador</b>	
<b>Descripción</b>	
<b>Pasos</b>	
<b>Requisitos probados</b>	
<b>Resultados esperados</b>	
<b>Resultados obtenidos</b>	

- **Identificador:** Título identificativo del caso de prueba, siguiendo el formato PS-XX, siendo XX un número secuencial.
- **Descripción:** breve explicación de la prueba que se va a llevar a cabo.
- **Pasos:** Secuencia de pasos que se deben realizar para pasar la prueba.
- **Requisitos probados:** Requisitos con los que se corresponde la prueba.
- **Resultados esperados:** Resultados que se espera obtener de la realización de la prueba.
- **Resultados obtenidos:** Resultados reales obtenidos de la realización de la prueba específica.



## 6.2 Pruebas llevadas a cabo

Aquí se detallan las pruebas que se han llevado a cabo para comprobar si el sistema cumple con los requisitos.

<b>Identificador</b>	PS-01
<b>Descripción</b>	La aplicación muestra una pantalla inicial con las distintas opciones que se pueden llevar a cabo.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación</li> <li>2. Comprobar que se ve una pantalla principal con las diferentes opciones.</li> </ol>
<b>Requisitos probados</b>	RF-01
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se carga la aplicación.</li> <li>• Se muestra una pantalla principal con la distintas alternativas.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 83: Prueba de sistema 1

<b>Identificador</b>	PS-02
<b>Descripción</b>	Se verán datos en la aplicación si hay un proyecto seleccionado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se despliega la aplicación.</li> <li>2. Se accede a la opción "Configuration" del menú principal.</li> <li>3. Se elige un proyecto de la lista de proyectos creados.</li> <li>4. Se accede a cualquier otra opción del proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-02
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• La aplicación muestra los datos cargados para el proyecto seleccionado.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 84: Prueba de sistema 2

<b>Identificador</b>	PS-03
<b>Descripción</b>	La aplicación estará vacía si no se selecciona ningún proyecto antes de acceder a las distintas pantallas.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se despliega la aplicación.</li> </ol>

	2. Se accede directamente a cualquiera de las opciones del menú, sin haber seleccionado ningún proyecto.
<b>Requisitos probados</b>	RF-02
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>No se muestran datos en ninguna de las pantallas de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 85: Prueba de sistema 3

<b>Identificador</b>	PS-04
<b>Descripción</b>	Añadir un nuevo proyecto con todos los datos introducidos.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>Desplegar la aplicación.</li> <li>Ir a la pantalla "Configuration".</li> <li>Añadir un nombre de proyecto.</li> <li>Añadir una descripción de proyecto</li> <li>Añadir el modelo de referencia.</li> <li>Aceptar.</li> </ol>
<b>Requisitos probados</b>	RF-03
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>Se crea correctamente el proyecto en la base de datos.</li> <li>Se muestra por pantalla un mensaje afirmativo.</li> <li>Aparece en la lista de proyectos el nuevo proyecto creado.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 86: Prueba de sistema 4

<b>Identificador</b>	PS-05
<b>Descripción</b>	Añadir un nuevo proyecto rellenando sólo los campos obligatorios. (nombre y modelo).
<b>Pasos</b>	<ol style="list-style-type: none"> <li>Desplegar la aplicación.</li> <li>Ir a la pantalla "Configuration".</li> <li>Añadir un nombre de proyecto.</li> <li>Añadir el modelo de referencia.</li> <li>Aceptar.</li> </ol>
<b>Requisitos probados</b>	RF-03
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>Se crea correctamente el proyecto en la base de datos.</li> </ul>

	<ul style="list-style-type: none"> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece el nuevo proyecto en la lista de proyectos disponibles.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 87: Prueba de sistema 5

<b>Identificador</b>	PS-06
<b>Descripción</b>	Añade un nuevo proyecto sin indicar el modelo de referencia.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration".</li> <li>3. Añadir un nombre de proyecto y descripción (opcional).</li> <li>4. Aceptar.</li> </ol>
<b>Requisitos probados</b>	RF-03
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ningún proyecto en la base de datos.</li> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• El proyecto no aparece en la lista de proyectos disponibles.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 88: Prueba de sistema 6

<b>Identificador</b>	PS-07
<b>Descripción</b>	Crear un "Artifact Definition" indicando todos los campos.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el <i>Reference Artifact</i> al que se le va a añadir.</li> <li>5. Pulsar "Add a new Artifact Definition".</li> <li>6. Añadir el nombre.</li> <li>7. Añadir la descripción.</li> <li>8. Pulsar "Submit"</li> </ol>
<b>Requisitos probados</b>	RF-04
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crea correctamente el "Artifact Definition" en la base de</li> </ul>

	<p>datos.</p> <ul style="list-style-type: none"> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece en la tabla de “Artifact Definitions” del “ReferenceArtefact” al que se le ha añadido.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 89: Prueba de sistema 7

<b>Identificador</b>	PS-08
<b>Descripción</b>	Crear un “Artifact Definition” dejando en blanco algún campo marcado como obligatorio.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Artifacts” del menú.</li> <li>4. Seleccionar el <i>Reference Artifact</i> al que se le va a añadir.</li> <li>5. Pulsar “Add a new Artifact Definition”.</li> <li>6. Añadir el nombre o la descripción (únicamente uno de ellos).</li> <li>7. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-04
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ningún “Artifact Definition” en la base de datos.</li> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• El “Artifact Definition” no aparece en la tabla del <i>Reference Artifact</i>.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 90: Prueba de sistema 8

<b>Identificador</b>	PS-09
<b>Descripción</b>	Crear un nuevo “Artifact” dentro de un “Artifact Definition” añadiendo todos los campos que se piden.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los</li> </ol>

	<p>creados.</p> <ol style="list-style-type: none"> <li>3. Acceder a la pantalla “Artifacts” del menú.</li> <li>4. Seleccionar el <i>Reference Artifact</i> al que se le va a añadir.</li> <li>5. Seleccionar el botón “See Artifacts” del “Artifact Definition” al que se va a añadir.</li> <li>6. Pulsar “Add a new Artifact”.</li> <li>7. Añadir el nombre del artefacto.</li> <li>8. Añadir su descripción.</li> <li>9. Añadir la versión del mismo.</li> <li>10. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-05
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crea correctamente el “Artifact” en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece en la tabla de “Artifacts” del “Artifact Definition” al que se le ha añadido.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 91: Prueba de sistema 9

<b>Identificador</b>	PS-10
<b>Descripción</b>	Crear un nuevo “Artifact” dentro de un “Artifact Definition” dejando algún campo de los marcados como obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Artifacts” del menú.</li> <li>4. Seleccionar el <i>Reference Artifact</i> al que se le va a añadir.</li> <li>5. Seleccionar el botón “See Artifacts” del “Artifact Definition” al que se va a añadir.</li> <li>6. Pulsar “Add a new Artifact”.</li> <li>7. Dejar en blanco, al menos, uno de los campos que aparecen (nombre, descripción o versión)</li> <li>8. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-05

<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ningún “Artifact” en la base de datos.</li> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• El “Artifact” no aparece en la tabla del “Artifact Definition” en el que se estaba intentando añadir.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 92: Prueba de sistema 10

<b>Identificador</b>	PS-11
<b>Descripción</b>	Modificar un “Artifact Definition” creado anteriormente, sin dejar ningún campo obligatorio en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Artifacts” del menú.</li> <li>4. Seleccionar el “Artifact Definition” que se quiere modificar.</li> <li>5. Pulsar en el botón “Edit”.</li> <li>6. Cambiar los datos que se quieran modificar.</li> <li>7. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-06
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifica el “Artifact Definition” concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Se ven los datos cambiados por pantalla.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 93: Prueba de sistema 11

<b>Identificador</b>	PS-12
<b>Descripción</b>	Modificar un “Artifact Definition” creado anteriormente, dejando en blanco alguno de los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Acceder a la pantalla “Artifacts” del menú.</li> <li>4. Seleccionar el “Artifact Definition” que se quiere modificar.</li> <li>5. Pulsar en el botón “Edit”.</li> <li>6. Cambiar los datos que se quieran modificar.</li> <li>7. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-06
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se modifica el “Artifact Definition” concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje de error, dándole la posibilidad al usuario de volver al formulario en el que se encontraba anteriormente..</li> <li>• Los datos referidos al “Artifact Definition” con el que está trabajando el usuario son los mismos que aparecían antes de intentar modificarlos.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 94: Prueba de sistema 12

<b>Identificador</b>	PS-13
<b>Descripción</b>	Modificar un “Artifact” sin dejar ninguno de los campos marcados como obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Artifacts” del menú.</li> <li>4. Seleccionar el “Artifact Definition” que se quiere modificar.</li> <li>5. Pulsar en el botón “See Artifacts”.</li> <li>6. Seleccionar el Artifact” que se quiere modificar.</li> <li>7. Pulsar el botón “Edit”.</li> <li>8. Cambiar los datos que se quieran modificar.</li> <li>9. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-07
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifica el “Artifact” concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> </ul>

	<ul style="list-style-type: none"> <li>• Se ven los datos cambiados por pantalla.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 95: Prueba de sistema 13

<b>Identificador</b>	PS-14
<b>Descripción</b>	Modificar un "Artifact" dejando alguno de los campos obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> <li>8. Cambiar los datos que se quieran modificar, dejando alguno de los marcados como obligatorios en blanco.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-07
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se modifica el "Artifact" concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje de error, dándole la posibilidad al usuario de volver al formulario en el que se encontraba anteriormente.</li> <li>• Los datos referidos al "Artifact" con el que está trabajando el usuario son los mismos que aparecían antes de intentar modificarlos.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 96: Prueba de sistema 14



<b>Identificador</b>	PS-15
<b>Descripción</b>	Añadir técnicas a un artefacto ya creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> <li>8. Marcar las técnicas que se quieren asociar al artefacto de las que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-19
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se añaden las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la lista de técnicas asociadas al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 97: Prueba de sistema 15

<b>Identificador</b>	PS-16
<b>Descripción</b>	Añadir roles a un artefacto ya creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> </ol>

	<ol style="list-style-type: none"> <li>8. Marcar los roles que se quieren asociar al artefacto de los que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-25
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se añaden las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la lista de roles asociados al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 98: Prueba de sistema 16

<b>Identificador</b>	PS-17
<b>Descripción</b>	Añadir requisitos a un artefacto ya creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> <li>8. Marcar los requisitos que se quieren asociar al artefacto de los que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-37
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se añaden las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la lista de requisitos asociados al mismo</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 99: Prueba de sistema 17

<b>Identificador</b>	PS-18
<b>Descripción</b>	Añadir atributos a un artefacto ya creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> <li>8. Marcar el grado de cumplimiento de los atributos que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-07
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se añaden las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la lista de propiedades y su grado de cumplimiento asociadas al mismo</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 100: Prueba de sistema 18

<b>Identificador</b>	PS-19
<b>Descripción</b>	Modificar las técnicas asociadas a un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> </ol>

	<ol style="list-style-type: none"> <li>8. Marcar las nuevas técnicas que se quieren asociar al artefacto de las que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-20
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se actualizan las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la nueva lista de técnicas asociadas al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 101: Prueba de sistema 19

<b>Identificador</b>	PS-20
<b>Descripción</b>	Modificar los roles asociados a un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> <li>8. Marcar los nuevos roles que se quieren asociar .</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-26
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se actualizan las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la nueva lista de roles asociados al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 102: Prueba de sistema 20

<b>Identificador</b>	PS-21
<b>Descripción</b>	Modificar los requisitos asociados a un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> <li>8. Marcar los nuevos requisitos que se quieren asociar al artefacto de los que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-38
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifican las relaciones correspondientes a esta modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la nueva lista de requisitos asociados al mismo</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 103: Prueba de sistema 21

<b>Identificador</b>	PS-22
<b>Descripción</b>	Modificar los atributos asociados a un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el Artifact" que se quiere modificar.</li> <li>7. Pulsar el botón "Edit".</li> </ol>

	<ol style="list-style-type: none"> <li>8. Marcar el grado de cumplimiento de los atributos que se le muestran al usuario.</li> <li>9. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-07
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifican las relaciones correspondientes a estar modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar los artefactos concretos, aparece, para cada uno de ellos, la lista de nuevas propiedades y su nuevo grado de cumplimiento asociadas al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 104: Prueba de sistema 22

<b>Identificador</b>	PS-23
<b>Descripción</b>	Borrar un "Artifact Definition".
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere borrar.</li> <li>5. Pulsar en el botón "Delete".</li> </ol>
<b>Requisitos probados</b>	RF-08
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se borra el "Artifact Definition" de la base de datos.</li> <li>• Se borran todos los artefactos referenciados al mismo de la base de datos.</li> <li>• Desaparece el "Artifact Definition" que se ha borrado de la vista de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 105: Prueba de sistema 23

<b>Identificador</b>	PS-24
<b>Descripción</b>	Borrar un artefacto previamente creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Artifacts" del menú.</li> <li>4. Seleccionar el "Artifact Definition" que se quiere modificar.</li> <li>5. Pulsar en el botón "See Artifacts".</li> <li>6. Seleccionar el "Artifact" que se quiere borrar.</li> <li>7. Pulsar el botón "Delete".</li> </ol>
<b>Requisitos probados</b>	RF-09
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se borra el artefacto de la base de datos.</li> <li>• Se borran todas las relaciones que existían con el resto de elementos de la base de datos.</li> <li>• Desaparece el artefacto que se ha borrado de la vista de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 106: Prueba de sistema 24

<b>Identificador</b>	PS-25
<b>Descripción</b>	Creación de una nueva técnica añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> <li>4. Acceder a la pantalla "Techniques" del menú lateral.</li> <li>5. Seleccionar la <i>Reference Technique</i> a la que se le va a añadir.</li> <li>6. Pulsar "Add a new Technique".</li> <li>7. Añadir el nombre.</li> <li>8. Añadir la descripción.</li> </ol>

	9. Pulsar "Submit"
<b>Requisitos probados</b>	RF-15
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crea correctamente la técnica en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece en la tabla de técnicas de la <i>Reference Technique</i> a la que se le ha añadido.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 107: Prueba de sistema 25

<b>Identificador</b>	PS-26
<b>Descripción</b>	Creación de una nueva técnica dejando en blanco alguno de los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> <li>4. Acceder a la pantalla "Techniques" del menú lateral.</li> <li>5. Seleccionar la <i>Reference Technique</i> a la que se le va a añadir.</li> <li>6. Pulsar "Add a new Technique".</li> <li>7. Pulsar "Submit"</li> </ol>
<b>Requisitos probados</b>	RF-15
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ninguna técnica nueva en la base de datos.</li> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• La nueva técnica no aparece en la tabla de la <i>Reference Technique</i>.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 108: Prueba de sistema 26



<b>Identificador</b>	PS-27
<b>Descripción</b>	Modificar una técnica ya creada añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> <li>4. Acceder a la pantalla "Techniques" del menú lateral.</li> <li>5. Seleccionar la técnica que se quiere modificar.</li> <li>6. Pulsar en el botón "Edit".</li> <li>7. Cambiar los datos que se quieran modificar.</li> <li>8. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-16
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifica la técnica concreta en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Se ven los datos cambiados por pantalla.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 109: Prueba de sistema 27

<b>Identificador</b>	PS-28
<b>Descripción</b>	Modificar una técnica ya creada dejando campos obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> <li>4. Acceder a la pantalla "Techniques" del menú lateral.</li> <li>5. Seleccionar la técnica que se quiere modificar.</li> <li>6. Pulsar en el botón "Edit".</li> <li>7. Cambiar los datos que se quieran modificar (dejando el nombre en blanco).</li> <li>8. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-16
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se modifica la técnica concreta en la base de datos.</li> </ul>

	<ul style="list-style-type: none"> <li>• Se muestra por pantalla un mensaje de error, dándole la posibilidad al usuario de volver al formulario en el que se encontraba anteriormente.</li> <li>• Los datos referidos a la técnica con la que está trabajando el usuario son los mismos que aparecían antes de intentar modificarlos.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 110: Prueba de sistema 28

<b>Identificador</b>	PS-29
<b>Descripción</b>	Borrar una técnica de las que el usuario ha creado previamente.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú superior.</li> <li>4. Acceder a la pantalla "Techniques" del menú lateral izquierdo.</li> <li>5. Seleccionar la técnica que se quiere borrar.</li> <li>6. Pulsar en el botón "Delete".</li> </ol>
<b>Requisitos probados</b>	RF-17
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se borra la técnica de la base de datos.</li> <li>• Se borran todas las relaciones que existían con el resto de elementos de la base de datos.</li> <li>• Desaparece la técnica que se ha borrado de la vista de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 111: Prueba de sistema 29

<b>Identificador</b>	PS-30
<b>Descripción</b>	Creación de un nuevo rol añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Roles” del menú lateral.</li> <li>5. Seleccionar el <i>Reference Role</i> al que se le va a añadir.</li> <li>6. Pulsar “Add a new Rol”.</li> <li>7. Añadir el nombre.</li> <li>8. Añadir la descripción.</li> <li>9. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-21
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crea correctamente el rol en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece en la tabla de roles del <i>Reference Role</i> al que se le ha añadido.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 112: Prueba de sistema 30

<b>Identificador</b>	PS-31
<b>Descripción</b>	Creación de un nuevo rol dejando en blanco alguno de los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Roles” del menú lateral.</li> <li>5. Seleccionar el <i>Reference Role</i> al que se le va a añadir.</li> <li>6. Pulsar “Add a new Participant”.</li> <li>7. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-21
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ningún rol nuevo en la base de datos.</li> </ul>

	<ul style="list-style-type: none"> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• El nuevo rol no aparece en la tabla del Reference Role.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 113: Prueba de sistema 31

<b>Identificador</b>	PS-32
<b>Descripción</b>	Modificar un rol ya creado añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Roles” del menú lateral.</li> <li>5. Seleccionar el rol que se quiere modificar.</li> <li>6. Pulsar en el botón “Edit”.</li> <li>7. Cambiar los datos que se quieran modificar.</li> <li>8. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-22
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifica el rol concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Se ven los datos cambiados por pantalla.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 114 : Prueba de sistema 32

<b>Identificador</b>	PS-33
<b>Descripción</b>	Modificar un rol ya creado dejando campos obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Roles” del menú lateral.</li> <li>5. Seleccionar el rol que se quiere modificar.</li> </ol>

	<ol style="list-style-type: none"> <li>Pulsar en el botón "Edit".</li> <li>Cambiar los datos que se quieran modificar (dejando el nombre en blanco).</li> <li>Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-22
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>No se modifica el rol concreto en la base de datos.</li> <li>Se muestra por pantalla un mensaje de error, dándole la posibilidad al usuario de volver al formulario en el que se encontraba anteriormente.</li> <li>Los datos referidos al rol con el que está trabajando el usuario son los mismos que aparecían antes de intentar modificarlos.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 115: Prueba de sistema 33

<b>Identificador</b>	PS-34
<b>Descripción</b>	Borrar un rol de los que el usuario ha creado previamente.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>Desplegar la aplicación.</li> <li>Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>Acceder a la pantalla "Processes" del menú superior.</li> <li>Acceder a la pantalla "Roles" del menú lateral izquierdo.</li> <li>Seleccionar el rol que se quiere borrar.</li> <li>Pulsar en el botón "Delete".</li> </ol>
<b>Requisitos probados</b>	RF-23
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>Se borra el rol de la base de datos.</li> <li>Se borran todas las relaciones que existían con el resto de elementos de la base de datos.</li> <li>Desaparece el rol que se ha borrado de la vista de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 116: Prueba de sistema 34

<b>Identificador</b>	PS-35
<b>Descripción</b>	Creación de una nueva actividad añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Activities” del menú lateral.</li> <li>5. Seleccionar la <i>Reference Activity</i> a la que se le va a añadir.</li> <li>6. Pulsar “Add a new Activity”.</li> <li>7. Añadir el nombre.</li> <li>8. Añadir la descripción.</li> <li>9. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-27
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crea correctamente la actividad en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece en la tabla de actividades de la <i>Reference Activity</i> a la que se le ha añadido.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 117: Prueba de sistema 35

<b>Identificador</b>	PS-36
<b>Descripción</b>	Creación de una nueva actividad dejando en blanco alguno de los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Activities” del menú lateral.</li> <li>5. Seleccionar la <i>Reference Activity</i> a la que se le va a añadir.</li> <li>6. Pulsar “Add a new Activity”.</li> <li>7. Pulsar “Submit”</li> </ol>
<b>Requisitos probados</b>	RF-27
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ninguna actividad nueva en la base de datos.</li> </ul>

	<ul style="list-style-type: none"> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• La nueva actividad no aparece en la tabla de la <i>Reference Activity</i>.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 118: Prueba de sistema 36

<b>Identificador</b>	PS-37
<b>Descripción</b>	Modificar una actividad ya creada añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> <li>4. Acceder a la pantalla "Activities" del menú lateral.</li> <li>5. Seleccionar la actividad que se quiere modificar.</li> <li>6. Pulsar en el botón "Edit".</li> <li>7. Cambiar los datos que se quieran modificar.</li> <li>8. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-28
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifica la actividad concreta en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Se ven los datos cambiados por pantalla.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 119: Prueba de sistema 37

<b>Identificador</b>	PS-38
<b>Descripción</b>	Modificar una actividad ya creada dejando campos obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Acceder a la pantalla “Activities” del menú lateral.</li> <li>5. Seleccionar la actividad que se quiere modificar.</li> <li>6. Pulsar en el botón “Edit”.</li> <li>7. Cambiar los datos que se quieran modificar (dejando el nombre en blanco).</li> <li>8. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-28
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se modifica la actividad concreta en la base de datos.</li> <li>• Se muestra por pantalla un mensaje de error, dándole la posibilidad al usuario de volver al formulario en el que se encontraba anteriormente.</li> <li>• Los datos referidos a la actividad con la que está trabajando el usuario son los mismos que aparecían antes de intentar modificarlos.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 120: Prueba de sistema 38

<b>Identificador</b>	PS-39
<b>Descripción</b>	Añadir productores a una actividad ya creada.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Activities” del menú lateral.</li> <li>5. Seleccionar la actividad que se quiere modificar.</li> <li>6. Pulsar en el botón “Edit”.</li> <li>7. Marcar los artefactos que se quieran asociar como productores de las actividades.</li> <li>8. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-31
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se añaden las relaciones correspondientes a estar modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar las actividades concretas, aparece,</li> </ul>



	para cada una de ellas, la lista de artefactos asociados al mismo.
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 121: Prueba de sistema 39

<b>Identificador</b>	PS-40
<b>Descripción</b>	Añadir usuarios a una actividad ya creada.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> <li>4. Acceder a la pantalla "Activities" del menú lateral.</li> <li>5. Seleccionar la actividad que se quiere modificar.</li> <li>6. Pulsar en el botón "Edit".</li> <li>7. Marcar los artefactos que se quieran asociar como usuarios de las actividades.</li> <li>8. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-31
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se añaden las relaciones correspondientes a estar modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar las actividades concretas, aparece, para cada una de ellas, la lista de artefactos asociados al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 122: Prueba de sistema 40

<b>Identificador</b>	PS-41
<b>Descripción</b>	Modificar los productores a una actividad ya creada.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Acceder a la pantalla “Activities” del menú lateral.</li> <li>5. Seleccionar la actividad que se quiere modificar.</li> <li>6. Pulsar en el botón “Edit”.</li> <li>7. Marcar los artefactos que se quieran asociar como productores de las actividades.</li> <li>8. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-32
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifican las relaciones correspondientes a estar modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar las actividades concretas, aparece, para cada una de ellas, la lista de artefactos asociados al mismo.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 123: Prueba de sistema 41

<b>Identificador</b>	PS-42
<b>Descripción</b>	Modificar los productores a una actividad ya creada.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Processes” del menú.</li> <li>4. Acceder a la pantalla “Activities” del menú lateral.</li> <li>5. Seleccionar la actividad que se quiere modificar.</li> <li>6. Pulsar en el botón “Edit”.</li> <li>7. Marcar los artefactos que se quieran asociar como usuarios de las actividades.</li> <li>8. Pulsar “Submit”.</li> </ol>
<b>Requisitos probados</b>	RF-32
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifican las relaciones correspondientes a estar modificación a la base de datos.</li> <li>• Se muestra un mensaje afirmativo por pantalla.</li> <li>• A la hora de visualizar las actividades concretas, aparece, para cada una de ellas, la lista de artefactos asociados al</li> </ul>

	mismo.
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 124: Prueba de sistema 42

<b>Identificador</b>	PS-43
<b>Descripción</b>	Borrar una actividad de las que el usuario ha creado previamente.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Processes" del menú superior.</li> <li>4. Acceder a la pantalla "Activities" del menú lateral izquierdo.</li> <li>5. Seleccionar la actividad que se quiere borrar.</li> <li>6. Pulsar en el botón "Delete".</li> </ol>
<b>Requisitos probados</b>	RF-29
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se borra la actividad de la base de datos.</li> <li>• Se borran todas las relaciones que existían con el resto de elementos de la base de datos.</li> <li>• Desaparece la actividad que se ha borrado de la vista de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 125: Prueba de sistema 43

<b>Identificador</b>	PS-44
<b>Descripción</b>	Creación de un nuevo requisito añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Justification" del menú.</li> <li>4. Seleccionar el "Reference Requirement" al que se le va a añadir.</li> <li>5. Pulsar "Add a new Requirement".</li> </ol>

	6. Añadir el nombre. 7. Añadir la descripción. 8. Pulsar "Submit"
<b>Requisitos probados</b>	RF-33
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crea correctamente el requisito en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Aparece en la tabla de roles del "Reference Requirement" al que se le ha añadido.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 126: Prueba de sistema 44

<b>Identificador</b>	PS-45
<b>Descripción</b>	Creación de un nuevo requisito dejando en blanco alguno de los campos marcados como obligatorios.
<b>Pasos</b>	1. Desplegar la aplicación. 2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados. 3. Acceder a la pantalla "Justification" del menú. 4. Seleccionar el "Reference Requirement" al que se le va a añadir. 5. Pulsar "Add a new Requirement". 6. Pulsar "Submit"
<b>Requisitos probados</b>	RF-33
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se crea ningún requisito nuevo en la base de datos.</li> <li>• Se redirige al usuario a una página de error y se le da la opción de volver al formulario que estaba rellenando.</li> <li>• El nuevo requisito no aparece en la tabla del "Reference Requirement".</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 127: Prueba de sistema 45

<b>Identificador</b>	PS-46
<b>Descripción</b>	Modificar un requisito ya creado añadiendo todos los campos marcados como obligatorios.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Justification" del menú.</li> <li>4. Seleccionar el requisito que se quiere modificar.</li> <li>5. Pulsar en el botón "Edit".</li> <li>6. Cambiar los datos que se quieran modificar.</li> <li>7. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-34
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifica el requisito concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje afirmativo.</li> <li>• Se ven los datos cambiados por pantalla.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 128: Prueba de sistema 46

<b>Identificador</b>	PS-47
<b>Descripción</b>	Modificar un requisito ya creado dejando campos obligatorios en blanco.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla "Configuration" y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla "Justification" del menú.</li> <li>4. Seleccionar el requisito que se quiere modificar.</li> <li>5. Pulsar en el botón "Edit".</li> <li>6. Cambiar los datos que se quieran modificar (dejando el nombre en blanco).</li> <li>7. Pulsar "Submit".</li> </ol>
<b>Requisitos probados</b>	RF-34
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• No se modifica el requisito concreto en la base de datos.</li> <li>• Se muestra por pantalla un mensaje de error, dándole la</li> </ul>

	<p>posibilidad al usuario de volver al formulario en el que se encontraba anteriormente.</p> <ul style="list-style-type: none"> <li>Los datos referidos al requisito con el que está trabajando el usuario son los mismos que aparecían antes de intentar modificarlos.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 129: Prueba de sistema 47

<b>Identificador</b>	PS-48
<b>Descripción</b>	Borrar un requisito de los que el usuario ha creado previamente.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Justification” del menú superior.</li> <li>4. Seleccionar el requisito que se quiere borrar.</li> <li>5. Pulsar en el botón “Delete”.</li> </ol>
<b>Requisitos probados</b>	RF-35
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>Se borra el requisito de la base de datos.</li> <li>Se borran todas las relaciones que existían con el resto de elementos de la base de datos.</li> <li>Desaparece el requisito que se ha borrado de la vista de la aplicación.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 130: Prueba de sistema 48

<b>Identificador</b>	PS-49
<b>Descripción</b>	Ver el grado de avance del proyecto con el que se está trabajando cuando hay seleccionado un proyecto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Ir a la pantalla “Configuration” y elegir un proyecto de los creados.</li> <li>3. Acceder a la pantalla “Project” del menú.</li> </ol>
<b>Requisitos probados</b>	RF-39

<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Los gráficos se cargan de forma dinámica.</li> <li>• Se ve el porcentaje de referencias utilizadas en el proyecto.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 131: Prueba de sistema 49

<b>Identificador</b>	PS-50
<b>Descripción</b>	Ver el grado de avance del proyecto sin seleccionar previamente un proyecto de la lista de proyectos en curso.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Desplegar la aplicación.</li> <li>2. Acceder a la pantalla “Project” del menú principal.</li> </ol>
<b>Requisitos probados</b>	RF-39
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• La pantalla aparece vacía.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 132: Prueba de sistema 50

<b>Identificador</b>	PS-51
<b>Descripción</b>	Crear un <i>Reference Assurance Framework</i>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Crear un nuevo diagrama.</li> <li>3. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-40
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crean los elementos.</li> <li>• Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 133: Prueba de sistema 51

<b>Identificador</b>	PS-52
<b>Descripción</b>	Modificar un elemento del <i>Reference Assurance Framework</i> que se ha creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Seleccionar el <i>Reference Assurance Framework</i> sobre el que se quieren hacer las modificaciones.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Localizar el elemento a cambiar.</li> <li>4. Cambiar los datos necesarios.</li> <li>5. Guardar el proyecto</li> </ol>
<b>Requisitos probados</b>	RF-41
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se modifican los elementos.</li> <li>• Se ven los cambios en el diagrama o árbol (según corresponda).</li> <li>• Se modifica el fichero de texto plano creado.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 134: Prueba de sistema 52

<b>Identificador</b>	PS-53
<b>Descripción</b>	Borrar un <i>Reference Assurance Framework</i> creado.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el <i>Reference Assurance Framework</i> que se quiera borrar.</li> <li>3. Borrarlo.</li> </ol>
<b>Requisitos probados</b>	RF-49
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se borra el <i>Reference Assurance Framework</i>.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 135: Prueba de sistema 53

<b>Identificador</b>	PS-54
<b>Descripción</b>	Añadir un evento a un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el Artifact Definition que contiene el artefacto al que se le quiere añadir.</li> <li>3. Pulsar “see artifacts”.</li> <li>4. Pulsar “edit”.</li> <li>5. Pulsar “add event”</li> <li>6. Añadir los datos correspondientes</li> <li>7. Pulsar “submit”.</li> </ol>
<b>Requisitos probados</b>	RF-12



<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Crea un nuevo evento en la base de datos.</li> <li>• Se muestra un nuevo evento en la lista de eventos asociados al artefacto.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 136: Prueba de sistema 54

<b>Identificador</b>	PS-55
<b>Descripción</b>	Seleccionar tipo de evento
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el Artifact Definition que contiene el artefacto al que se le quiere añadir.</li> <li>3. Pulsar “see artifacts”.</li> <li>4. Pulsar “edit”.</li> <li>5. Pulsar “add event”</li> <li>6. Seleccionar el tipo de evento que se va a añadir.</li> </ol>
<b>Requisitos probados</b>	RF-13
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Aparece un menú desplegado según el tipo de evento que se haya seleccionado.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 137: Prueba de sistema 55

<b>Identificador</b>	PS-56
<b>Descripción</b>	Añadir una evaluación a un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el Artifact Definition que contiene el artefacto al que se le quiere añadir.</li> <li>3. Pulsar “see artifacts”.</li> <li>4. Pulsar “edit”.</li> <li>5. Pulsar “add event”</li> <li>6. Seleccionar “Evaluation” entre los tipos de eventos.</li> <li>7. Añadir los datos correspondientes</li> <li>8. Pulsar “submit”.</li> </ol>
<b>Requisitos probados</b>	RF-14

<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Crea un nuevo evento en la base de datos.</li> <li>• Crea una nueva evaluación en la base de datos.</li> <li>• Se muestra un nuevo evento en la lista de eventos asociados al artefacto.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 138: Prueba de sistema 56

<b>Identificador</b>	PS-57
<b>Descripción</b>	Ver detalles de un <i>Artifact Definition</i> .
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el Artifact Definition.</li> <li>3. Pulsar “see”</li> </ol>
<b>Requisitos probados</b>	RF-10
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se muestran por pantalla todos los atributos o datos de un <i>Artefact Definition</i>.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 139: Prueba de sistema 57

<b>Identificador</b>	PS-58
<b>Descripción</b>	Ver detalles de un artefacto.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el Artifact Definition que contiene el artefacto que se quiere ver.</li> <li>3. Pulsar “see artifacts”</li> <li>4. Seleccionar el artefacto que se quiere visualizar.</li> <li>5. Pulsar “see details”.</li> </ol>
<b>Requisitos probados</b>	RF-11
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se muestran por pantalla todos los datos del artefacto.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 140: prueba de sistema 58

<b>Identificador</b>	PS-59
<b>Descripción</b>	Ver detalles de una técnica.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar la técnica que se quiere ver.</li> <li>3. Pulsar “see”</li> </ol>
<b>Requisitos probados</b>	RF-18
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se muestran por pantalla todos los datos de la técnica.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 141: Prueba de sistema 59

<b>Identificador</b>	PS-60
<b>Descripción</b>	Ver detalles de un rol.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar el rol que se quiere ver.</li> <li>3. Pulsar “see”</li> </ol>
<b>Requisitos probados</b>	RF-24
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se muestran por pantalla todos los datos del rol.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 142: Prueba de sistema 60

<b>Identificador</b>	PS-61
<b>Descripción</b>	Ver detalles de una actividad.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Seleccionar la actividad que se quiere ver.</li> <li>3. Pulsar “see”</li> </ol>
<b>Requisitos probados</b>	RF-30
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se muestran por pantalla todos los datos de la actividad.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 143: Prueba de sistema 61.

<b>Identificador</b>	PS-62
<b>Descripción</b>	Ver detalles de un requisito.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>4. Abrir la aplicación.</li> <li>5. Seleccionar el requisito que se quiere ver.</li> <li>6. Pulsar “see”</li> </ol>
<b>Requisitos probados</b>	RF-36
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se muestran por pantalla todos los datos del requisito.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 144: Prueba de sistema 62

<b>Identificador</b>	PS-63
<b>Descripción</b>	Añadir nuevos <i>Reference Artifacts</i> al modelo
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Crear un nuevo diagrama.</li> <li>3. Añadir los artefactos necesarios al diagrama.</li> <li>4. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-42
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crean los elementos.</li> <li>• Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 145: Prueba de sistema 63

<b>Identificador</b>	PS-64
<b>Descripción</b>	Añadir nuevos <i>Reference Techniques</i> al modelo
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Abrir el modelo creado.</li> <li>3. Añadir una nueva técnica al árbol.</li> <li>4. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-43
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crean los elementos.</li> <li>• Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 146: Prueba de sistema 64

<b>Identificador</b>	PS-65
<b>Descripción</b>	Añadir nuevos <i>Reference Roles</i> al modelo
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Abrir el modelo creado.</li> <li>3. Añadir un nuevo rol al árbol.</li> <li>4. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-44
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crean los elementos.</li> <li>• Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 147: Prueba de sistema 65

<b>Identificador</b>	PS-66
<b>Descripción</b>	Añadir nuevos <i>Reference Activities</i> al modelo
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Abrir el modelo creado.</li> <li>3. Añadir una nueva actividad al árbol.</li> <li>4. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-45
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crean los elementos.</li> <li>• Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 148: Prueba de sistema 66

<b>Identificador</b>	PS-67
<b>Descripción</b>	Añadir nuevos <i>Reference Requirements</i> al modelo
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Abrir el modelo creado.</li> <li>3. Añadir un nuevo requisito al árbol.</li> <li>4. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-46
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>• Se crean los elementos.</li> </ul>

	<ul style="list-style-type: none"> <li>Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 149: Prueba de sistema 67

<b>Identificador</b>	PS-68
<b>Descripción</b>	Añadir nuevos <i>Reference Attributes</i> a los <i>Reference Artefacts</i> del modelo
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Abrir el modelo creado.</li> <li>3. Añadir un nuevo rol al árbol.</li> <li>4. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-47
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>Se crean los elementos.</li> <li>Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 150: Prueba de sistema 68

<b>Identificador</b>	PS-69
<b>Descripción</b>	Añadir nuevas relaciones entre los <i>Reference Artifacts</i> del modelo.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Abrir la herramienta.</li> <li>2. Abrir el modelo creado.</li> <li>3. Seleccionar en el diagrama el artefacto de origen.</li> <li>4. Seleccionar en el diagrama el artefacto de destino.</li> <li>5. Guardar el proyecto.</li> </ol>
<b>Requisitos probados</b>	RF-48
<b>Resultados esperados</b>	<ul style="list-style-type: none"> <li>Se crean los elementos.</li> <li>Se crea el fichero .raf correspondiente.</li> </ul>
<b>Resultados obtenidos</b>	Prueba pasada.

Tabla 151: Prueba de sistema 69

### 6.3 Estado de las pruebas de sistema y trazabilidad

Una vez expuestas las distintas pruebas de sistema que se han llevado a cabo, en este punto se mostrará de forma sencilla el estado final de las pruebas realizadas. Aparecerá, por cada prueba, si ha sido un éxito o un fracaso.

En la misma tabla, se van a indicar también los requisitos a los que hace referencia cada caso de prueba, con el fin de demostrar que se han probado todos los requisitos funcionales explicados en la parte de análisis de la memoria.

Identificador de la prueba	Requisito funcional asociado	Descripción de la prueba	Estado
PS-01	RF-01	La aplicación muestra una pantalla inicial con las distintas opciones que se pueden llevar a cabo.	Éxito ✓
PS-02	RF-02	Se verán datos en la aplicación si hay un proyecto seleccionado.	Éxito ✓
PS-03	RF-02	La aplicación estará vacía si no se selecciona ningún proyecto antes de acceder a las distintas pantallas.	Éxito ✓
PS-04	RF-03	Añadir un nuevo proyecto con todos los datos introducidos.	Éxito ✓
PS-05	RF-03	Añadir un nuevo proyecto rellenando sólo los campos obligatorios. (nombre y modelo).	Éxito ✓
PS-06	RF-03	Añade un nuevo proyecto sin indicar el modelo de referencia.	Éxito ✓
PS-07	RF-04	Crear un "Artifact Definition" indicando todos los campos.	Éxito ✓
PS-08	RF-04	Crear un "Artifact Definition" dejando en blanco algún campo marcado como obligatorio.	Éxito ✓
PS-09	RF-05	Crear un nuevo "Artifact" dentro de un "Artifact Definition" añadiendo todos los campos que se piden.	Éxito ✓
PS-10	RF-05	Crear un nuevo "Artifact" dentro de un "Artifact Definition" dejando algún campo de los marcados como obligatorios en blanco.	Éxito ✓

PS-11	RF-06	Modificar un "Artifact Definition" creado anteriormente, sin dejar ningún campo obligatorio en blanco.	Éxito ✓
PS-12	RF-06	Modificar un "Artifact Definition" creado anteriormente, dejando en blanco alguno de los campos marcados como obligatorios.	Éxito ✓
PS-13	RF-07	Modificar un "Artifact" sin dejar ninguno de los campos marcados como obligatorios en blanco.	Éxito ✓
PS-14	RF-07	Modificar un "Artifact" dejando alguno de los campos obligatorios en blanco.	Éxito ✓
PS-15	RF-19	Añadir técnicas a un artefacto ya creado.	Éxito ✓
PS-16	RF-25	Añadir roles a un artefacto ya creado.	Éxito ✓
PS-17	RF-37	Añadir requisitos a un artefacto ya creado.	Éxito ✓
PS-18	RF-07	Añadir atributos a un artefacto ya creado.	Éxito ✓
PS-19	RF-20	Modificar las técnicas asociadas a un artefacto.	Éxito ✓
PS-20	RF-26	Modificar los roles asociados a un artefacto.	Éxito ✓
PS-21	RF-38	Modificar los requisitos asociados a un artefacto.	Éxito ✓
PS-22	RF-07	Modificar los atributos asociados a un artefacto.	Éxito ✓
PS-23	RF-08	Borrar un "Artifact Definition".	Éxito ✓
PS-24	RF-09	Borrar un artefacto previamente creado.	Éxito ✓
PS-25	RF-15	Creación de una nueva técnica añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-26	RF-15	Creación de una nueva técnica dejando en blanco alguno de los campos marcados como obligatorios.	Éxito ✓
PS-27	RF-16	Modificar una técnica ya creada añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-28	RF-16	Modificar una técnica ya creada dejando campos obligatorios en blanco.	Éxito ✓
PS-29	RF-17	Borrar una técnica de las que el usuario ha creado previamente.	Éxito ✓
PS-30	RF-21	Creación de un nuevo rol añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-31	RF-21	Creación de un nuevo rol dejando en blanco alguno de	Éxito ✓



		los campos marcados como obligatorios.	
PS-32	RF-22	Modificar un rol ya creado añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-33	RF-22	Modificar un rol ya creado dejando campos obligatorios en blanco.	Éxito ✓
PS-34	RF-23	Borrar un rol de los que el usuario ha creado previamente.	Éxito ✓
PS-35	RF-27	Creación de una nueva actividad añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-36	RF-27	Creación de una nueva actividad dejando en blanco alguno de los campos marcados como obligatorios.	Éxito ✓
PS-37	RF-28	Modificar una actividad ya creada añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-38	RF-28	Modificar una actividad ya creada dejando campos obligatorios en blanco.	Éxito ✓
PS-39	RF-31	Añadir productores a una actividad ya creada.	Éxito ✓
PS-40	RF-31	Añadir usuarios a una actividad ya creada.	Éxito ✓
PS-41	RF-32	Modificar los productores a una actividad ya creada.	Éxito ✓
PS-42	RF-32	Modificar los productores a una actividad ya creada.	Éxito ✓
PS-43	RF-29	Borrar una actividad de las que el usuario ha creado previamente.	Éxito ✓
PS-44	RF-33	Creación de un nuevo requisito añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-45	RF-33	Creación de un nuevo requisito dejando en blanco alguno de los campos marcados como obligatorios.	Éxito ✓
PS-46	RF-34	Modificar un requisito ya creado añadiendo todos los campos marcados como obligatorios.	Éxito ✓
PS-47	RF-34	Modificar un rol ya creado dejando campos obligatorios en blanco.	Éxito ✓
PS-48	RF-35	Borrar un requisito de los que el usuario ha creado previamente.	Éxito ✓
PS-49	RF-39	Ver el grado de avance del proyecto con el que se está trabajando cuando hay seleccionado un proyecto.	Éxito ✓
PS-50	RF-39	Ver el grado de avance del proyecto sin seleccionar	Éxito ✓

		previamente un proyecto de la lista de proyectos en curso.	
PS-51	RF-40	Crear un <i>Reference Assurance Framework</i>	Éxito ✓
PS-52	RF-41	Modificar un elemento del <i>Reference Assurance Framework</i> que se ha creado.	Éxito ✓
PS-53	RF-49	Borrar un <i>Reference Assurance Framework</i> creado.	Éxito ✓
PS-54	RF-12	Añadir un evento a un artefacto.	Éxito ✓
PS-55	RF-13	Seleccionar tipo de evento	Éxito ✓
PS-56	RF-14	Añadir una evaluación a un artefacto.	Éxito ✓
PS-57	RF-10	Ver detalles de un <i>Artifact Definition</i> .	Éxito ✓
PS-58	RF-11	Ver detalles de un artefacto.	Éxito ✓
PS-59	RF-18	Ver detalles de una técnica.	Éxito ✓
PS-60	RF-24	Ver detalles de un rol.	Éxito ✓
PS-61	RF-30	Ver detalles de una actividad.	Éxito ✓
PS-62	RF-36	Ver detalles de un requisito.	Éxito ✓
PS-63	RF-42	Añadir nuevos <i>Reference Artifacts</i> al modelo	Éxito ✓
PS-64	RF-43	Añadir nuevos <i>Reference Techniques</i> al modelo	Éxito ✓
PS-65	RF-44	Añadir nuevos <i>Reference Roles</i> al modelo	Éxito ✓
PS-66	RF-45	Añadir nuevos <i>Reference Activities</i> al modelo	Éxito ✓
PS-67	RF-46	Añadir nuevos <i>Reference Requirements</i> al modelo	Éxito ✓
PS-68	RF-47	Añadir nuevos <i>Reference Attributes</i> a los <i>Reference Artefacts</i> del modelo	Éxito ✓
PS-69	RF-48	Añadir nuevas relaciones entre los <i>Reference Artifacts</i> del modelo.	Éxito ✓

Tabla 152: Trazabilidad entre pruebas de sistema y requisitos funcionales.

## 7. Gestión del Proyecto

En este apartado se expondrán la planificación que se ha seguido para la realización del trabajo y el presupuesto estimado del mismo, teniendo en cuenta el personal que ha participado, los recursos necesarios y su duración total.

### 7.1 Planificación

La realización del proyecto comenzó a principios de enero y finalizó a principios de junio, haciendo un total de 5 meses. Durante ese tiempo, las horas de trabajo diarias no fueron constantes, sino que dependieron de la carga de trabajo de cada fase del trabajo.

El proyecto se ha dividido en tres fases, que se detallan a continuación:

- **Fase 1:** Esta fase se desarrolló entre los meses de enero y febrero. En ella, se pusieron en común los conocimientos, y se estudiaron las distintas alternativas posibles para la realización de la herramienta. Una vez hecho lo anterior, se procedió a la fase de análisis de la aplicación, definiendo los requisitos que debería cumplir y los casos de uso a implementar.
- **Fase 2:** Una vez terminada la primera fase, en esta se implementó la herramienta, teniendo en cuenta todas las decisiones tomadas en la primera fase. La implementación de la herramienta se explica en el punto 4. Diseño e Implementación. Esta fase tiene lugar durante los meses de marzo, abril y principios de mayo.
- **Fase 3:** Por último, se llevaron a cabo todas las pruebas finales de la herramienta, con el fin de corregir posibles fallos que no se habían tenido en cuenta en la segunda fase. Además, se completó la memoria del trabajo, explicando todo el trabajo realizado.

En la siguiente tabla se exponen las diferentes tareas llevadas a cabo en cada fase de forma más explícita, mostrando también las horas de trabajo que ha supuesto cada fase:

Fase	Tareas	Horas	Horas acumuladas
Fase 1: Análisis	<ul style="list-style-type: none"> <li>Investigación de las herramientas.</li> <li>Creación del meta-modelo.</li> <li>Toma de decisiones de almacenamiento e implementación.</li> <li>Definición de requisitos y casos de uso.</li> </ul>	60	60
Fase 2: Implementación	<ul style="list-style-type: none"> <li>Implementación de los casos de uso definidos durante el análisis de la herramienta.</li> <li>Modificaciones necesarias en el análisis para el funcionamiento de la aplicación.</li> </ul>	300	360
Fase 3: Pruebas finales	<ul style="list-style-type: none"> <li>Pruebas finales.</li> <li>Documentación.</li> </ul>	100	<b>460</b>

Tabla 153: Resumen de las fases del proyecto.

### 7.1.1 Diagrama de Gantt

A continuación se expone el diagrama de Gantt asociado al proyecto. En él, se han representado las tareas asociadas a cada una de las fases que se han explicado en la tabla anterior, con el tiempo de trabajo estimado para cada una, en días.

La primera de las imágenes mostrará los siguientes datos:

- Señal que indica si la tarea se ha finalizado o sigue en curso. Todas las tareas se han finalizado.
- Modo: planificación automática o manual de la tarea en cuestión. En este caso, todas las tareas se han planificado de forma manual.
- Nombre: Tarea que se va a representar. Cada una de las fases principales tiene las sub tareas que se han indicado asociadas.

- Duración: Tiempo estimado de trabajo en cada una de las tareas. Contabiliza únicamente los días laborables del calendario en curso.
- Comienzo: fecha de inicio de la realización de la tarea.
- Fin: fecha de finalización de la tarea en cuestión.
- Predecesoras: Identificador de las tareas anteriores que es necesario haber realizado y finalizado para iniciarla.

	i	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	✓	✈	<b>Fase 1: Análisis</b>	36 días	<b>lun 11/01/16</b>	<b>lun 29/02/16</b>	
2	✓	✈	Estudio de las herramientas	11 días	lun 11/01/16	lun 25/01/16	
3	✓	✈	Creación del metamodelo	11 días	lun 18/01/16	lun 01/02/16	
4	✓	✈	Estudio de las alternativas	11 días	lun 18/01/16	lun 01/02/16	
5	✓	✈	Definición de casos de uso	5 días	mar 02/02/16	lun 08/02/16	2;3;4
6	✓	✈	Definición de requisitos	15 días	mar 09/02/16	lun 29/02/16	5
7	✓	✈	<b>Fase 2: Implementación</b>	55 días	<b>mar 01/03/16</b>	<b>lun 16/05/16</b>	<b>1</b>
8	✓	✈	Implementación de los casos de uso	55 días	mar 01/03/16	lun 16/05/16	
9	✓	✈	<b>Fase 3: Pruebas finales</b>	22 días	<b>mar 17/05/16</b>	<b>mié 15/06/16</b>	<b>1;7</b>
10	✓	✈	Pruebas finales	22 días	mar 17/05/16	mié 15/06/16	
11	✓	✈	Documentación final	22 días	mar 17/05/16	mié 15/06/16	

Ilustración 67: Planificación de las tareas del proyecto.

El siguiente diagrama muestra el curso de las tareas indicadas en la imagen anterior.



Ilustración 68: Diagrama de Gantt

## 7.2 Presupuesto

En esta sección se detallan los costes asociados a la elaboración del proyecto. Se diferenciará entre costes de mano de obra o personal, y coste asociado a los materiales utilizados.

### 7.2.1 Coste de Personal

En este punto se recogen los costes asociados al personal que ha participado en la realización del proyecto.

La siguiente tabla recoge los datos de cada participante, así como su coste por horas, el tiempo dedicado al proyecto y el coste final asociado a cada miembro. Se han tenido en cuenta los costes totales de las personas, no haciendo distinción entre las distintas partidas que se asocian a ese coste( sueldos y salarios y seguridad social del negocio).

Para calcular el coste por hora de cada trabajador se han tenido en cuenta los salarios medios de personal en una empresa consultora. De esta forma, los costes correspondientes a sueldos y salarios serían los siguientes:

- Ingeniero Junior: 20.000 euros anuales brutos.
- Ingeniero Senior: 30.000 euros anuales brutos.

Nombre	Cargo	Coste por hora (en euros)	Horas dedicadas	Coste asociado (en euros)	final
Elena Correás Montiel	Ingeniero Junior	11,00	400	4400,00	
José Luís de la Vara González	Ingeniero Senior	16,00	60	960,00	
<b>TOTAL</b>				<b>5360,00 euros</b>	

Tabla 154: Coste de personal del proyecto.

### 7.2.2 Coste de Material

En este apartado se tienen en cuenta todos aquellos costes imputados al proyecto relacionados con el material que se ha utilizado durante el desarrollo del mismo.

Para realizar la estimación de los costes materiales, solo se han tenido en cuenta aquellos que han supuesto un coste económico, no incluyéndose aquellos programas o recursos de distribución libre.

En la siguiente tabla se pueden diferenciar los siguientes apartados para cada recurso material utilizado:

- Coste total: expresado en unidades monetarias (euros).
- Amortización: Periodo de tiempo durante el cual se deprecia el valor de un bien. En este caso, se utilizará una amortización lineal, en la que cada la degradación del recurso será la misma cada mes. Para exponer este dato, se han tenido en cuenta los datos publicados por las *tablas oficiales de años y porcentajes de amortización para sociedades a partir de 2015*. (Cuéntica, 2015)
- Tiempo de uso: periodo de la vida del recurso concreto que ha sido utilizado para el proyecto.
- Dedicación al proyecto: Porcentaje del tiempo de uso dedicado exclusivamente al proyecto.
- Coste final asociado: se calculará mediante la siguiente fórmula:

$$\text{Coste Final} = \frac{\text{Coste Total}}{\text{Amortización}} * \text{Tiempo de uso} * \text{Dedicación al Proyecto}$$

La siguiente tabla muestra los costes asociados a cada recurso material utilizado:

Recurso	Coste total (en euros)	Amortización (en meses)	Tiempo de uso (en meses)	Dedicación al proyecto	Coste final asociado (en euros)
MacBook Pro 13"	1449	60	6	80%	115,92
Microsoft Office	149,95	36	6	100%	24,99
<b>TOTAL</b>					<b>140,91</b>

Tabla 155: Coste material del proyecto.

### 7.2.3 Coste total asociado al proyecto

El coste total del proyecto será la suma de todos los costes expuestos en los apartados anteriores. Se ha considerado no necesario añadir otros costes indirectos, como pueden ser los derivados del consumo eléctrico necesario para realizar el proyecto.

El coste total será por tanto el siguiente:

Descripción	Cuantía
Coste de personal	5360,00
Coste de material	140,91
<b>TOTAL</b>	<b>5500,91</b>

Tabla 156: Coste final del proyecto.



## 8. Conclusiones y Líneas Futuras

Este último apartado del documento recoge las conclusiones obtenidas durante la realización del proyecto y las posibles líneas futuras o mejoras de la aplicación para incrementar sus funcionalidades.

### 8.1 Conclusiones

En este punto se explican las conclusiones obtenidas de la realización del proyecto, tanto desde el punto de vista personal como desde el punto de vista objetivo de la consecución de los objetivos marcados al principio del mismo y explicados en la introducción del documento.

Desde el punto de vista personal, este proyecto ha servido, por un lado, para aprender sobre el desarrollo de software a través de modelos y conocer el entorno en el que se desarrollan este tipo de herramientas. Por el otro lado, el hecho de tener que realizar una aplicación completa y desde cero, y pasando por todas las fases del desarrollo, ha servido para completar la formación obtenida durante los años de carrera en las distintas asignaturas cursadas, y aprender a combinar lo aprendido en cada una de ellas. La realización de un proyecto como este desde cero no sólo hace que se completen los conocimientos adquiridos durante la carrera, sino que también se aprende a crecer como trabajador, a gestionar el tiempo con el que se cuenta y a adquirir conocimientos de manera autosuficiente.

Desde el punto de vista objetivo, se han conseguido los objetivos planteados al principio del proyecto, lo que hace estar satisfecho con el trabajo realizado. Dichos objetivos eran, y se consiguieron, de la siguiente forma:

- **Creación de un editor gráfico de modelos**, mediante la herramienta GMF de Eclipse, permitiendo al usuario crear un nuevo *Reference Assurance Framework* para un estándar específico cada vez que lo necesite para un nuevo proyecto.
- **Implementación de una herramienta que permita añadir evidencias según lo establecido en el modelo creado**, a través de la aplicación diseñada para el manejo de evidencias, en la que se pueden gestionar los distintos elementos que forman el modelo (artefactos, requisitos, técnicas, roles y actividades).

- **Monitorización del progreso del proyecto** que el usuario esté llevando a cabo, a través de la aplicación que se ha desarrollado, de la forma explicada en los puntos anteriores, en la que se puede comprobar en todo momento la completitud o avance de cada elemento de referencia del modelo.
- **Gestión del ciclo de vida de las evidencias**, a través de la aplicación implementada, añadiendo las distintas versiones de los artefactos a cada definición, señalando cual es la versión anterior del nuevo artefacto creado, y añadiendo eventos a los mismos para su control.

## 8.2 Líneas Futuras

El proyecto realizado es una primera implementación de un posible futuro proyecto que podrá abarcar una serie de funcionalidades que no se han tenido en cuenta para el desarrollo de éste. Estas son:

- **Generación de informes en forma de documento:** Exportación de los datos introducidos por el usuario durante el desarrollo del proyecto que esté llevando a cabo en forma de documento, de forma que quede toda la información de las evidencias recogidas en un mismo archivo.
- **Mejora de la interfaz:** Nuevas posibles formas de representación de la información mediante pantalla. Por ejemplo, en el caso de la presentación de los elementos, hacerlo en forma de lista escalonada, en vez de en tablas, para que de cada elemento padre se cree una nueva lista de elementos hijos. De esta forma, la navegación entre páginas se reduciría.
- **Editor multi-diagrama para los *Reference Assurance Framework*:** con editores gráficos también para el resto de elementos del modelo, ya que en el caso que se ha desarrollado sólo se permitía la edición gráfica de los artefactos y sus relaciones y atributos.
- **Soporte a más actividades del ciclo de vida de las evidencias:** con funcionalidades como el análisis del impacto de los cambios, automatización de la gestión de la trazabilidad, visualización de dicha trazabilidad con modelos o matrices, importación de información desde otras herramientas que crean o gestionan

artefactos específicos, como podría ser el caso de, por ejemplo, herramientas de especificación de requisitos, diseño, modelado, *testing*, etc.

## 9. Bibliografía

- Altreonic. (s.f.). *GoedelWorks-Altreonic*. Obtenido de <http://www.altreonic.com/content/altreonic-approach-systems-engineering-goedelworks>
- Boletín Oficial del Estado*. (2015). Obtenido de *Boletín Oficial del Estado*: <https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750>
- Boletín Oficial del Estado*. (2015). Obtenido de *Boletín Oficial del Estado*: <https://www.boe.es/legislacion/codigos/codigo.php?id=67&modo=1&nota=0&tab=2>
- CE. (2016). Obtenido de [http://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/medical-devices/index\\_en.htm](http://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/medical-devices/index_en.htm)
- Check, P. I. (s.f.). *PTC Integrity Process Check*. Obtenido de PTC Integrity Process Check: [http://www.ptc.com/~media/Files/PDFs/ALM/Integrity/PTC-Integrity-Process\\_Check.ashx?la=en](http://www.ptc.com/~media/Files/PDFs/ALM/Integrity/PTC-Integrity-Process_Check.ashx?la=en)
- CRESO. (2011). Obtenido de <https://www.duo.uio.no/bitstream/handle/10852/8950/TorbjornSkybergKnutsen-master.pdf?sequence=2>
- Cuéntica. (2015). Obtenido de <https://ayuda.cuentica.com/anos-y-porcentaje-de-amortizacion-para-sociedades/>
- Director, P. I. (s.f.). *PTC Integrity Process Director*. Obtenido de PTC Integrity Process Director: [http://www.ptc.com/~media/Files/PDFs/ALM/Integrity/PTC\\_Integrity\\_Process\\_Director.ashx?la=en](http://www.ptc.com/~media/Files/PDFs/ALM/Integrity/PTC_Integrity_Process_Director.ashx?la=en)
- EASA. (2016). Obtenido de <https://www.easa.europa.eu>
- ERA. (2016). Obtenido de <http://www.era.europa.eu/Pages/Home.aspx>
- ESA. (2016). Obtenido de <http://www.esa.int/ESA>
- IEEE Computer Society. (s.f.). Obtenido de IEEE 830-1998: <http://standards.ieee.org/findstds/standard/830-1998.html>
- LDRA. (s.f.). Obtenido de *Cost-Effective Methodology for Achieving Automotive Software Compliance*: [https://vector.com/portal/medien/cmc/events/commercial\\_events/VU\\_Conference/VUC14/Cost-effective\\_methodology\\_for\\_achieving\\_automotive\\_software\\_compliance\\_v2.pdf](https://vector.com/portal/medien/cmc/events/commercial_events/VU_Conference/VUC14/Cost-effective_methodology_for_achieving_automotive_software_compliance_v2.pdf)
- Nair. (2014). *Characterization of Safety Evidence*.

NAIR. (s.f.). *Characterization of Safety Evidence for Assessment and Certification of Critical Systems*.

Nair, S. (2014). *Characterization of Safety Evidence*.

Nair, S. (2014). *Characterization of Safety Evidence for Assessment and Certification of Critical Systems*.

OPENCROSS. (s.f.). *OPENCROSS*. Obtenido de <http://www.opencross-project.eu>

RAE. (2016). Obtenido de <http://dle.rae.es/?id=H9WZ6YF>

Servlet-UC3M. (s.f.). *Laboratorio UC3M*. Obtenido de Laboratorio UC3M: <http://www.lab.inf.uc3m.es/~a0080802/RAI/servlet.html>

V.Delgado, I. (s.f.). Obtenido de <https://victdelr.wordpress.com>

WEFACT. (s.f.). *WEFACT*. Obtenido de WEFACT: <http://www.ait.ac.at/research-services/research-services-digital-safety-security/verification-and-validation/methods-and-tools/wefact-workflow-engine-for-analysis-certification-and-test/?L=1>